



Pedro Manuel Correia Almeida **Servidor de vídeo de elevada granularidade**



Pedro Manuel Correia Almeida Servidor de vídeo de elevada granularidade

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor Joaquim Arnaldo Martins, Professor Associado com Agregação do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro e do Professor Doutor Joaquim Sousa Pinto, Professor Auxiliar do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro.

o júri

presidente

Prof. Doutor José Luís Guimarães de Oliveira
professor Associado da Universidade de Aveiro

Prof. Doutor Joaquim Arnaldo Martins
professor Associado com Agregação da Universidade de Aveiro

Prof. Doutora Ana Alice Rodrigues Pereira Baptista
professora Auxiliar da Universidade do Minho

Prof. Doutor Joaquim Sousa Pinto
professor Auxiliar da Universidade de Aveiro

agradecimentos

Quero agradecer a todas as pessoas que contribuíram para que o presente trabalho fosse possível. Os meus especiais agradecimentos vão para os meus pais e os meus irmãos que sempre me apoiaram, para a Sandra Cruz, para os meus professores e orientadores que me ajudaram ao longo de todo o trabalho, para as pessoas da Assembleia da República que tornaram possível realizar este trabalho e finalmente para os meus colegas de mestrado e do curso de formação especializada com os quais criei novas amizades.

resumo

A presente dissertação propõe-se divulgar o desenvolvimento de um servidor de vídeo de elevada granularidade implementado sob a forma de um sistema de arquivo audiovisual. Descreve-se o trabalho desenvolvido na criação de um arquivo audiovisual que permite armazenar e indexar os conteúdos de vídeos respeitantes aos debates parlamentares da Assembleia da República Portuguesa. Utilizou-se o standard MPEG-7 como norma de descrição dos vídeos, uma base de dados XML para armazenar as descrições, criou-se uma aplicação de indexação de vídeo e um visualizador Web para permitir ver os vídeos através da Internet.

abstract

The present dissertation is considered to divulge the development of a high granularity video server implement in the form of an audiovisual archive system. It's described the work developed in the creation of an audiovisual archive that permits to store and index the content of the videos from the parliamentary debates of the Portuguese Parliament. The MPEG-7 standard was used as a norm to describe the video content, a XML database to store the descriptions, it was created a video indexing application and a web viewer to permit visualize the videos over the Internet.

Índice

1	INTRODUÇÃO	1
2	ENQUADRAMENTO	3
3	TECNOLOGIAS UTILIZADAS	5
3.1	XML	5
3.1.1	<i>Documentos XML Document-Centric</i>	<i>6</i>
3.1.2	<i>Documentos XML Data-Centric.....</i>	<i>7</i>
3.2	XML SCHEMAS.....	8
3.3	MPEG-7.....	10
3.4	BASE DE DADOS XML	11
3.5	WEB SERVICES	11
4	DESCRIÇÃO DOS VÍDEOS.....	13
4.1	INTRODUÇÃO	13
4.2	DESCRIÇÃO DE CONTEÚDO AUDIOVISUAL COM MPEG-7.....	13
4.2.1	<i>Elementos MPEG-7</i>	<i>14</i>
4.2.2	<i>Componentes MPEG-7.....</i>	<i>15</i>
4.2.3	<i>MDS - Multimedia Description Schemes.....</i>	<i>15</i>
4.2.4	<i>Modelo de descrição.....</i>	<i>17</i>
4.2.5	<i>Exemplo de uma descrição MPEG-7.....</i>	<i>28</i>
4.3	DATA BINDING	33
4.3.1	<i>Introdução</i>	<i>33</i>
4.3.2	<i>Arquitectura JAXB (Java Architecture for XML Binding).....</i>	<i>33</i>
4.3.3	<i>Processo de Data Binding no JAXB.</i>	<i>35</i>
4.3.4	<i>Descrição da estrutura do XML Schema.....</i>	<i>36</i>
5	ARQUITECTURA DO ARQUIVO AUDIVISUAL.....	45
5.1	CAMADA DE DADOS	45
5.1.1	<i>Servidor de vídeos</i>	<i>46</i>
5.1.2	<i>Base de dados de Intervenções.....</i>	<i>47</i>
5.1.3	<i>Base de dados XML com descrições dos vídeos.....</i>	<i>48</i>

5.2	CAMADA LÓGICA.....	50
5.2.1	<i>Ligação à base de dados de intervenções</i>	50
5.2.2	<i>Ligação à base de dados XML</i>	51
5.3	CAMADA DE APRESENTAÇÃO.....	54
6	APLICAÇÃO DE INDEXAÇÃO DE VÍDEO	55
6.1	LINGUAGEM DE PROGRAMAÇÃO UTILIZADA	55
6.2	INTERFACE	55
6.2.1	<i>Janela de tempo</i>	56
6.2.2	<i>Janela de visualização do vídeo</i>	56
6.2.3	<i>Janela de intervenções</i>	57
6.2.4	<i>Janela de anotações</i>	58
6.2.5	<i>Menu de acções e menu de configuração</i>	59
6.3	ARQUITECTURA DA APLICAÇÃO DE INDEXAÇÃO DE VÍDEO	60
6.3.1	<i>Ligação à base de dados XML</i>	60
6.4	PROCESSO DE CRIAÇÃO DE UMA DESCRIÇÃO VÍDEO	63
7	VISUALIZADOR WEB.....	71
7.1	INTRODUÇÃO	71
7.2	MODO DE FUNCIONAMENTO	71
7.3	VISUALIZAÇÃO DE UM VÍDEO	73
8	DESEMPENHO DA BASE DE DADOS XML XINDICE.....	75
8.1	INTRODUÇÃO	75
8.2	METODOLOGIA	75
8.3	APRESENTAÇÃO DOS RESULTADOS.....	77
8.3.1	<i>Acesso através do driver XML:DB API</i>	77
8.3.2	<i>Acesso com Web Services</i>	95
8.3.3	<i>Análise das pesquisas</i>	98
9	CONCLUSÕES E TRABALHO FUTURO	103
	REFERÊNCIAS	105
	BIBLIOGRAFIA	109

ANEXO A.....	111
UTILIZAÇÃO DA FERRAMENTA JAXB	111
PROCEDIMENTO DE COMPILAÇÃO DO <i>XML SCHEMA</i> :	111

Índice de Figuras

figura 1 - Documento <i>XML Document-Centric</i>	6
figura 2 - Documento <i>XML Data-Centric</i>	7
figura 3 - Exemplo de um <i>XML Schema</i>	9
figura 4 - Exemplo de documento <i>XML</i> com <i>XML Schema</i>	10
figura 5 - Representação das relações entre os elementos do standard <i>MPEG-7</i> [MARTINEZ, 2002].....	14
figura 6 - Organização dos elementos MDS [MARTINEZ, 2002]	16
figura 7 - Modelo <i>MPEG-7</i> utilizado	19
figura 8 - Descritor <i>MPEG-7 MediaFormat</i>	21
figura 9 - Descrição <i>XML</i> com o descritor <i>MediaFormat</i>	22
figura 10 - Descritor <i>MPEG-7 MediaInstance</i>	23
figura 11 - Descrição <i>XML</i> com o descritor <i>MediaInstance</i>	23
figura 12 - Descritor <i>MPEG-7 AudioVisualSegment</i>	24
figura 13 - Descrição <i>XML</i> com o descritor <i>AudioVisualSegment</i>	25
figura 14 - Descritor <i>MPEG-7 VideoSegment</i>	26
figura 15 - Descrição <i>XML</i> com o descritor <i>VideoSegment</i>	26
figura 16 - Descritor <i>MPEG-7 TextAnnotation</i>	27
figura 17 - Descrição <i>XML</i> com o descritor <i>TextAnnotation</i>	28
figura 18 - Exemplo de uma descrição de um conteúdo audiovisual	32
figura 19 - Arquitectura da ferramenta <i>JAXB</i> [SUN, 2003b]	34
figura 20 - Passos gerais do processo de <i>Data Binding</i> [SUN, 2003b].....	36
figura 21 - Arquitectura do sistema de arquivo audiovisual.....	45
figura 22 - Organização dos registos na base de dados <i>XML</i>	49
figura 23 - Métodos disponibilizados pelo <i>Web Service</i> de acesso à base de dados <i>XML</i> ..	53
figura 24 - Interface da aplicação de indexação de vídeo.....	55
figura 25 - Janela de tempo	56
figura 26 - Janela de visualização do vídeo	57
figura 27 - Janela de intervenções	58
figura 28 - Janela de anotações com resumo de uma intervenção.....	59
figura 29 - Menu de acções	59

figura 30 - Configurações da aplicação	59
figura 31 - Diagrama de um documento <i>WSDL</i> [ORACLE,2003].....	61
figura 32 - Descrição <i>WSDL</i> do <i>Web Service</i>	62
figura 33 - Propriedades de um vídeo.....	64
figura 34 - Descrição <i>XML</i> inicial do vídeo	65
figura 35 - Criação de uma descrição	66
figura 36 - Janela de anotações da intervenção do deputado Francisco Louçã.....	67
figura 37 - Janela de anotações com o classificador do tempo.....	67
figura 38 - Registo de notas numa intervenção	68
figura 39 - Registo de descritores numa intervenção	69
figura 40 - Esquema de funcionamento do <i>Web Service</i>	71
figura 41 - Referência <i>Web</i> para o <i>Web Service</i> xmldbws	72
figura 42 - <i>Link</i> para seleccionar a opção de visualizar o vídeo da sessão.....	73
figura 43 - Interface <i>Web</i> de visualização dos vídeos	74
figura 44 - Interface da aplicação de análise	76
figura 45 - Gráficos de tempo despendido ao adicionar 50.000 documentos.....	79
figura 46 - Gráficos do tempo despendido ao adicionar 40.000 documentos	80
figura 47 - Gráficos do tempo despendido ao adicionar 30.000 documentos	81
figura 48 - Gráficos do tempo despendido ao adicionar 20.000 documentos	82
figura 49 - Gráfico do tempo despendido ao adicionar 10.000 documentos.....	83
figura 50 - Gráficos do tempo despendido ao obter 50.000 documentos	85
figura 51 - Gráficos do tempo despendido ao obter 40.000 documentos	86
figura 52 - Gráficos do tempo despendido ao obter 30.000 documentos	87
figura 53 - Gráficos do tempo despendido ao obter 20.000 documentos	88
figura 54 - Gráfico do tempo despendido ao obter 10.000 documentos.....	89
figura 55 - Gráficos do tempo despendido ao remover 50.000 documentos.....	91
figura 56 - Gráficos do tempo despendido ao remover 40.000 documentos.....	92
figura 57 - Gráficos do tempo despendido ao remover 30.000 documentos.....	93
figura 58 - Gráficos do tempo despendido ao remover 20.000 documentos.....	94
figura 59 - Gráfico do tempo despendido ao remover 10.000 documentos	95
figura 60 - Gráficos do tempo despendido com <i>Web Services</i>	97
figura 61 - Explorador do projecto Video	113

Índice de Tabelas

Tabela 1 - Tabela de codificação de vídeo	46
Tabela 2 - Descrição dos elementos de um documento <i>WSDL</i> [ORACLE, 2003]	61
Tabela 3 - Tempos registados ao adicionar 50.000 documentos	79
Tabela 4 - Tempos registados ao adicionar 40.000 documentos	80
Tabela 5 - Tempos registados ao adicionar 30.000 documentos	81
Tabela 6 - Tempos registados ao adicionar 20.000 documentos	82
Tabela 7 - Tempos registados ao adicionar 10.000 documentos	83
Tabela 8 - Tempos registados ao obter 50.000 documentos.....	85
Tabela 9 - Tempos registados ao obter 40.000 documentos.....	86
Tabela 10 - Tempos registados ao obter 30.000 documentos.....	87
Tabela 11 - Tempos registados ao obter 20.000 documentos.....	88
Tabela 12 - Tempos registados ao obter 10.000 documentos.....	89
Tabela 13 - Tempos registados ao remover 50.000 documentos	91
Tabela 14 - Tempos registados ao remover 40.000 documentos	92
Tabela 15 - Tempos registados ao remover 30.000 documentos	94
Tabela 16 - Tempos registados ao remover 20.000 documentos	94
Tabela 17 - Tempos registados ao remover 10.000 documentos	95
Tabela 18 - Tempos registados com <i>Web Services</i>	97
Tabela 19 - Tempos de pesquisa por descritores em 50.000 documentos.....	98
Tabela 20 - Tempos de pesquisa por ID em 50.000 documentos.....	98
Tabela 21 - Características da base de dados XML com 50.000 documentos	98
Tabela 22 - Tempos de pesquisa por descritores em 40.000 documentos.....	99
Tabela 23 - Tempos de pesquisa por ID em 40.000 documentos.....	99
Tabela 24 - Características da base de dados XML com 40.000 documentos	99
Tabela 25 - Tempos de pesquisa por descritores em 30.000 documentos.....	99
Tabela 26 - Tempos de pesquisa por ID em 30.000 documentos.....	99
Tabela 27 - Características da base de dados XML com 30.000 documentos	100
Tabela 28 - Tempos de pesquisa por descritores em 20.000 documentos.....	100
Tabela 29 - Tempos de pesquisa por ID em 20.000 documentos.....	100
Tabela 30 - Características da base de dados XML com 20.000 documentos	100

Tabela 31 - Tempos de pesquisa por descritores em 10.000 documentos.....	100
Tabela 32 - Tempos de pesquisa por ID em 10.000 documentos.....	101
Tabela 33 - Características da base de dados XML com 10.000 documentos	101

1 INTRODUÇÃO

Esta dissertação descreve o estudo que foi desenvolvido para obter o grau de Mestre em Engenharia Electrónica e Telecomunicações.

A dissertação tem como tema “Servidor de vídeo de elevada granularidade”. O objectivo do trabalho realizado é avaliar as tecnologias actuais que permitiram a construção do mesmo, implementado sob a forma de um arquivo audiovisual.

No decorrer do trabalho constatou-se que ainda existem muitos poucos sistemas que permitam caracterizar e armazenar informação relativa a conteúdos audiovisuais. Os poucos sistemas de arquivo audiovisual existentes até ao momento [Ryu, 2002] são trabalhos científicos e não estão disponíveis no mercado. A maior parte dos produtos comerciais existentes de indexação permitem indexar e armazenar informação relativa a documentos de texto, imagens, mas, quando se trata de armazenar informação relativa a um conteúdo audiovisual, os sistemas actuais pouco mais permitem do que armazenar sumários, resumos ou alguns descritores relativos a todo o conteúdo dos mesmos.

O servidor desenvolvido tem como objectivo armazenar informação audiovisual, indexar a mesma e permitir efectuar pesquisas de modo a obter apenas segmentos de vídeo que digam respeito aos descritores utilizados na pesquisa.

Um sistema com estas características vem ao encontro de uma das necessidades actuais do Arquivo Histórico e Parlamentar da Assembleia da República Portuguesa, uma vez que este pretende construir um sistema de arquivo audiovisual. Este trabalho foi realizado em cooperação com os mesmos, inserido num conjunto de projectos desenvolvidos anteriormente e associado, principalmente, ao sistema dos Diários da Assembleia da República Electrónicos [PINTO, 2001a].

Inicialmente, a dissertação faz um resumo do estado actual da arte no tocante às tecnologias sobre as quais assenta o sistema de informação, nomeadamente o *XML*, *XML Schemas*, *MPEG-7*, *Base de Dados XML*, e *Web Services*.

Apresentam-se depois os elementos pertencentes ao *standard MPEG-7* que se considerou serem fundamentais para descrever o conteúdo audiovisual e um modelo para caracterizar conteúdo audiovisual baseado em elementos *MPEG-7*. Este modelo permite caracterizar

detalhadamente o conteúdo audiovisual de um vídeo pertencente a uma Sessão Parlamentar da Assembleia da República.

Seguidamente apresenta-se a arquitectura do sistema de informação desenvolvido, bem como as suas características.

Apresenta-se também a aplicação de indexação de vídeo que foi desenvolvida para o sistema de arquivo audiovisual. A aplicação permite a múltiplos utilizadores indexarem simultaneamente diferentes vídeos das várias sessões parlamentares. Serve como *interface* que permite aos utilizadores visualizar, alterar, eliminar e adicionar informação a uma base de dados *XML* que contém as descrições dos vídeos.

Apresenta-se a *interface Web* que permite visualizar remotamente os segmentos de vídeo. Esta *interface* é apresentada inserida no sistema dos Diários da Assembleia da República Electrónicos, surgindo como uma extensão ao sistema original.

Apresenta-se o estudo elaborado para analisar o desempenho da base de dados *XML XIndice*. Os testes incidiram sobre os tempos de pesquisa e tempos de acesso a documentos.

Finalmente apresentam-se as conclusões do trabalho e sugestões para trabalho futuro.

2 ENQUADRAMENTO

No geral esta dissertação descreve todo o estudo e desenvolvimento que foi realizado na criação de um arquivo audiovisual que permita armazenar e indexar os conteúdos de vídeos respeitantes aos debates parlamentares da Assembleia da República Portuguesa.

Este arquivo surge na sequência de um conjunto de projectos anteriores que estão a ser desenvolvidos pela Universidade de Aveiro para a Assembleia da República Portuguesa. O Arquivo Parlamentar da Assembleia da República está a desenvolver um conjunto de sistemas que permitem gerir os vários conteúdos da sua Biblioteca Digital. Estes encontram-se distribuídos pelo Arquivo Histórico Parlamentar, Arquivo Fotográfico Parlamentar e Diários da Assembleia da República electrónicos.

Neste momento alguns desses sistemas já se encontram em fase avançada de implementação, como por exemplo os Diários da Assembleia da República electrónicos [PINTO, 2001b] e o Arquivo Histórico Parlamentar, e outros estão ainda em fase de estudo e desenvolvimento.

Associado principalmente ao sistema dos Diários da Assembleia da República electrónicos, o sistema desenvolvido no âmbito desta tese tem como objectivo permitir a visualização de um vídeo completo de uma sessão dos debates parlamentares Portugueses, ou de segmentos de vídeo da mesma, que correspondam às intervenções dos vários oradores que vão participando no decorrer da sessão. Como exemplo, é utilizado o vídeo de uma sessão parlamentar da Assembleia da República com duas horas de duração. Ao longo da sessão são debatidos vários temas. Alguns deles poderão ser de interesse para uma pessoa em particular. Em vez de o utilizador visualizar o vídeo completo e procurar pelos instantes em que, por exemplo, um determinado orador se encontra a efectuar uma intervenção sobre um tema em particular, este sistema tem como objectivo permitir apresentar de imediato a informação que o utilizador deseja, mostrando-lhe somente, vamos supor, os dois minutos em que um determinado orador se encontra a discursar.

Para que tal seja possível é necessário caracterizar um filme de vídeo pertencente a uma Sessão Parlamentar da Assembleia da República, decompor o vídeo em vários segmentos e caracterizar devidamente os segmentos de modo a ser possível extrair informações sobre aqueles que correspondem às várias intervenções parlamentares.

3 TECNOLOGIAS UTILIZADAS

Ao longo da criação deste sistema de arquivo audiovisual foram utilizadas várias tecnologias que estão por excelência ligadas à criação de sistemas de informação. Teve-se sempre em consideração que o sistema devia ser construído de uma forma distribuída permitindo suportar vários utilizadores em simultâneo.

Descrevem-se de seguida as principais tecnologias sobre as quais assenta o sistema. Alguns detalhes vão ser mais aprofundados à medida que se descreve o sistema implementado.

3.1 XML

O *XML*, *eXtensible Markup Language*, [W3C, 2000a] é uma recomendação do *World Wide Web Consortium* (W3C) e surge como uma evolução do *SGML*, *Standard Generalized Markup Language* [ISO, 2001], que é uma linguagem baseada em marcadores. Tem por objectivo responder a algumas limitações do *HTML*, *HyperText Markup Language* [W3C, 1999a]. Rapidamente se tornou evidente que as constantes alterações que eram feitas ao *HTML*, tendo em conta as necessidades dos utilizadores da *Internet*, não eram suficientes e daí a necessidade da criação de uma nova linguagem, baseada em marcadores, mais robusta e extensível. [W3C, 2000a]

O *XML* surge então como uma linguagem de marcadores que permite relacionar o conteúdo de texto com os marcadores pelas quais o mesmo se encontra delimitado.

Uma das mais importantes diferenças entre o *XML* e o *HTML* é que enquanto no *HTML* os marcadores estão todos definidos e a sua sintaxe é conhecida, no *XML* podem-se criar os nossos próprios marcadores com uma sintaxe e semântica específicas, daí a grande extensibilidade desta linguagem de marcas.

Um documento *XML* é um documento de texto. Qualquer máquina, desde que tenha conhecimento da semântica do conteúdo do documento, tem a possibilidade de interpretar correctamente o documento, independentemente do sistema operativo que utilize. Essa característica faz com que o *XML* seja bastante utilizado como uma linguagem de

comunicação entre aplicações. Existem dois tipos de documentos XML: *Document-Centric*, e *Data-Centric* que se apresentam em seguida. [BOURETT, 2003]

3.1.1 Documentos XML *Document-Centric*

Convém neste ponto clarificar a distinção entre os dois tipos de documentos XML que se podem encontrar. Os documentos XML *Document-Centric* são utilizados para descrever informação para consumo humano. Estes documentos são caracterizados pela sua estrutura bastante irregular e pelo seu conteúdo misto, o que aumenta a complexidade do documento. A ordem pela qual aparecem os elementos dentro do documento é completamente irrelevante.

A figura 1 apresenta um exemplo de um documento XML *Document-Centric* com a descrição de uma viagem. Começa com uma pequena introdução, passando depois pela descrição da viagem e termina com uma lista com o itinerário seguido.

```
<?xml version="1.0" encoding="ISO-8859-1">
<Viagem>
  <Introducao>
    A viagem a <Destino>Espanha</Destino> promovida pela
    empresa <Empresa> Férias de Sonho </Empresa> é <Resumo>uma
    grande viagem , apesar de ser aqui tão perto</Resumo>.
  </Introducao>
  <Descricao>
    <Paragrafo>
      Esta viagem a Espanha pretende mostrar um pouco da
      cultura e do modo de viver deste nosso povo vizinho.
      Vamos visitar algumas cidades entre as quais
      <cidade>Salamanca</cidade> e <cidade>Bilbao</cidade>
      e vamos poder observar desde monumentos históricos
      como a <destaque>Catedral de Salamanca</destaque > a
      monumentos em homenagem à arte contemporânea como o
      <destaque >Museu de Guggenheim</destaque >
    </Paragrafo>
    <Paragrafo>Itinerário</Paragrafo>
    <Lista>
      <item>Aveiro - Partida da Estação de Comboios</item>
      <item>Salamanca - Visita à Catedral e almoço </item>
      <item>Bilbao - visita ao museu de Guggenheim</item>
    </Lista>
  </Descricao>
</Viagem>
```

figura 1 - Documento XML *Document-Centric*

Constata-se facilmente que esta estrutura é bastante irregular e que um documento deste tipo tem como principal objectivo distinguir secções de texto dentro de um documento e não caracterizar informação de um modo estruturado.

3.1.2 Documentos *XML Data-Centric*

O tipo de documento *XML* mais usual é o documento *XML Data-Centric*. Estes documentos usam o *XML* como linguagem de descrição e normalmente são utilizados como fonte de informação para outras aplicações. São caracterizados pela sua estrutura regular e pelo seu conteúdo simples. Normalmente a ordem pela qual aparecem os elementos é muito importante.

A figura 2 apresenta um exemplo de um documento *XML Data-Centric* com informação sobre tipos de cafés, contendo o seu nome e o seu preço.

```
<?xml version="1.0" encoding="ISO-8859-1">
<priceList>
  <coffee>
    <name>Mocha Java</name>
    <price>11.95</price>
  </coffee>
  <coffee>
    <name>Sumatra</name>
    <price>12.50</price>
  </coffee>
</priceList>
```

figura 2 - Documento *XML Data-Centric*

As marcas *XML* delimitam o conteúdo e fornecem informação sobre a estrutura dos dados. Devido à sua estrutura homogénea é possível utilizar este tipo de documentos para arquivar e pesquisar informação.

Neste caso o documento *XML* apresentado tem um elemento principal com o nome *priceList*. Este elemento contém dois elementos *coffee* e cada um deles vai conter dois elementos, um com o nome *name* e outro com o nome *price*. Para o primeiro elemento *coffee* dentro do elemento *priceList*, pode-se verificar que o elemento *name* contém o valor

“*Mocha Java*” e o elemento *price* contém o valor “11.95” e para o segundo elemento *coffee* que aparece dentro do elemento *priceList* pode-se verificar que o elemento *name* contém o valor “*Sumatra*” e o elemento *price* contém o valor “12.50”. Intuitivamente chega-se à conclusão que este documento *XML* contém uma listagem com preços de cafés e que cada elemento *coffee* representa uma marca de café. Os elementos *name* e *price* representam respectivamente o nome e o preço de cada um dos cafés.

A informação apresentada neste tipo de documentos surge devidamente estruturada tornando-os mais apropriados para serem utilizados em sistemas de armazenamento de dados.

3.2 XML SCHEMAS

Apesar de um documento *XML* apresentar os dados delimitados por marcadores, nada impede que um utilizador interprete os dados à sua maneira não tendo em conta a semântica dos mesmos. Daí surge a necessidade de uma linguagem que permita descrever a estrutura de um documento *XML*.

Surgem inicialmente os *DTD's* (*Document Type Definition*) [W3C, 2000b] propostos pelo W3C como forma de definir uma estrutura para os documentos *XML*.

Mais tarde e devido a algumas limitações dos *DTD's*, surgem os *XML Schemas* [W3C, 2001a] como recomendação do W3C.

O objectivo de um *XML Schema* é garantir que se pode construir um documento *XML* de acordo com uma determinada estrutura. Os *XML Schemas* permitem definir os elementos e atributos que podem aparecer dentro de um documento *XML*, as posições onde aparecem os elementos e atributos, a ordem dos elementos filhos, o número de elementos filhos, se um elemento pode estar vazio ou não, os tipos de dados para os elementos e atributos, os valores por defeito para elementos e atributos, etc.

Algumas vantagens dos *XML Schemas* em relação aos *DTD's* são as seguintes [W3SCHOLLS, 2003]

- São mais ricos e mais práticos que os *DTD's*
- Suportam mais tipos de dados
- Utilizam sintaxe *XML*
- Suportam *namespaces*

Por exemplo, para o caso anterior de um documento *XML* com o lista de preços de cafés poderíamos definir o seguinte *XML Schema*:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
<xs:element name="priceList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="coffee" type="coffeeType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="coffeeType">
  <xs:sequence>
    <xs:element ref="name"/>
    <xs:element ref="price"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="name" type="xs:string"/>
<xs:element name="price" type="xs:decimal"/>
</xs:schema>
```

figura 3 - Exemplo de um *XML Schema*

Analisando o *XML Schema* apresentado na figura 3 pode-se concluir que o elemento *priceList* é composto por uma sequência de elementos *coffee* que são de um tipo complexo com o nome *coffeeType*. Os elementos do tipo *coffeeType* são constituídos por uma sequência de dois elementos: um elemento *name* que é uma variável do tipo *string* e um elemento *price* que é uma variável do tipo *decimal*. São estas características que definem inequivocamente quais os tipos de valores que os elementos devem conter e por que ordem devem aparecer dentro de um documento *XML*.

A análise deste *XML Schema* favorece também a interoperabilidade entre aplicações, pois os *XML Schemas* definem os tipos de dados para os elementos de um documento *XML* em particular, permitindo assim a uma aplicação que receba o documento *XML* e que tenha conhecimento do *XML Schema*, a possibilidade de interpretar correctamente os valores que os elementos contêm.

Normalmente, dentro de um documento *XML* faz-se uma referência ao *XML Schema* que é utilizado para definir as características do documento *XML* em particular. O documento *XML* que se apresenta na figura 4 tem a sua estrutura definida por um *XML Schema* que se encontra na seguinte localização <http://www.ieeta.pt/SchemaCoffee.xsd>.

```
<?xml version="1.0" encoding="UTF-8">
<priceList xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="
http://www.ieeta.pt/SchemaCoffee.xsd">
  <coffee>
    <name>Mocha Java</name>
    <price>11.95</price>
  </coffee>
  <coffee>
    <name>Sumatra</name>
    <price>12.50</price>
  </coffee>
</priceList>
```

figura 4 - Exemplo de documento *XML* com *XML Schema*

Deste modo já se tem informação dentro do documento *XML* que permite saber qual a localização do *XML Schema* com o qual se pode validar e interpretar a informação contida no mesmo.

3.3 MPEG-7

Com a banalização da Internet em banda larga e com a disponibilização de ferramentas para codificar e aceder a sistemas de informação audiovisual, é também cada vez mais comum aceder-se a sistemas que disponibilizam informação multimédia. Esta informação pode estar representada em vários formatos de média, como por exemplo, imagens, gráficos, modelos 3D, áudio, vídeo, etc. Contudo, inicialmente esta informação era para ser consumida directamente pelo utilizador humano, começam também agora a surgir os primeiros sistemas computacionais que criam, trocam, disponibilizam e reutilizam a informação audiovisual. [MARTINEZ, 2002]

O *MPEG-7* é um *standard* desenvolvido pelo grupo de trabalho *MPEG (Moving Picture Experts Group)* que permite efectuar a descrição *standardizada* de vários tipos de informação multimédia. O objectivo deste *standard* é permitir uma procura rápida e eficiente de material audiovisual que é de especial interesse para um utilizador em particular. Este *standard* não abrange a área de extracção automática de descritores nem

especifica um motor de busca que possa utilizar os descritores, permitindo assim aos fabricantes de *software* construírem as suas próprias ferramentas, aumentando assim a competitividade e as funcionalidades destes produtos.

O *standard MPEG-7* usa o *XML* e os *XML Schemas* como linguagem descritiva, permitindo assim facilidades de extensibilidade e de utilização. Este factor permite também aumentar a interoperabilidade, não estando dependente de uma plataforma ou de um fabricante de *software*. [MARTINEZ, 2002]

3.4 BASE DE DADOS XML

A base de dados que armazena informação sobre o conteúdo audiovisual é uma base de dados nativa *XML*. O *SGBD (Sistema de Gestão de Base de Dados)* que se utiliza tem por nome *XIndice* [APACHE, 2003a] e é uma plataforma *open-source* desenvolvida pela *Apache*. [APACHE, 2003a]

O factor que levou a optar pela utilização de uma base de dados *XML* nativa foi o facto de as descrições do conteúdo do vídeo serem armazenadas em documentos *XML*.

Deste modo tira-se partido das funcionalidades associadas às bases de dados *XML* no armazenamento e pesquisa de dados *XML*.

3.5 WEB SERVICES

Os *Web Services* [W3C, 2002] são serviços oferecidos através da *Web* [ARMSTRONG, 2003]. O principal objectivo de utilização dos *Web Services* é disponibilizar uma plataforma que permita a comunicação entre a base de dados *XML* e as aplicações clientes de um modo transparente, garantindo desta forma uma melhor modularidade do sistema.

Este tipo de abordagem permite mais tarde utilizar outros *SGBD's* sem ser necessário recompilar ou reconstruir a aplicação. Cria-se assim um nível de abstracção entre a base de dados *XML* e as aplicações clientes.

As principais funções de acesso à base de dados *XML*, como por exemplo: obter documentos, retirar documentos e inserir documentos, são disponibilizadas às aplicações clientes através de *Web Services*.

4 DESCRIÇÃO DOS VÍDEOS

4.1 INTRODUÇÃO

O objectivo que se pretende alcançar na construção do sistema de arquivo audiovisual é caracterizar um filme de vídeo pertencente a uma Sessão Parlamentar da Assembleia da República, decompor o vídeo em vários segmentos e caracterizar devidamente os segmentos de vídeo de modo a ser possível extrair informações sobre as intervenções parlamentares existentes no mesmo.

O primeiro passo realizado na concepção do sistema consistia na elaboração de um modelo, baseado no *standard MPEG-7*, para a descrição do conteúdo audiovisual. Após este passo, foi definido um *XML Schema* que permite descrever a estrutura dos documentos *XML* com descritores *MPEG-7*, onde ficam armazenadas as descrições dos vídeos. Mais tarde, a partir deste *XML Schema*, foram geradas um conjunto de classes *JAVA* que permitiram construir a aplicação de indexação de vídeo.

Neste capítulo apresenta-se um modelo com elementos *MPEG-7* que permite caracterizar detalhadamente um conteúdo audiovisual de um vídeo pertencente a uma Sessão Parlamentar da Assembleia da República e aprofundam-se um pouco os conceitos utilizados no mesmo para uma melhor compreensão das funcionalidades que lhe estão associadas.

4.2 DESCRIÇÃO DE CONTEÚDO AUDIOVISUAL COM MPEG-7

Descreve-se em seguida o estudo efectuado para obter o modelo baseado no *standard MPEG-7* que permite descrever o conteúdo audiovisual dos vídeos pertencentes à colecção de vídeos da Assembleia da República com os debates parlamentares.

Descreve-se resumidamente o *standard MPEG-7* para se obter uma imagem conceptual das funcionalidades associadas ao mesmo.

4.2.1 Elementos *MPEG-7*

O *standard MPEG-7* é constituído por três elementos que permitem efectuar as descrições de conteúdo audiovisual. Estes elementos são os seguintes [MARTINEZ, 2002]:

1. *Descriptors* (D) – Representações de características, definem a sintaxe e a semântica de cada representação para cada característica.
2. *Description Schemes* (DS) - Especificam a estrutura e a semântica das relações entre componentes. Estes componentes podem ser quer *Descriptors* quer *Description Schemes*.
3. *Description Definition Language* (DDL) – Permite a criação de novos *Description Schemes* e *Descriptors*. Permite também a extensão ou a modificação de *Description Schemes* existentes.

A figura 5 permite obter uma melhor compreensão do modo como se interligam os elementos que constituem o *standard MPEG-7*.

Os mecanismos da *Description Definition Language* que permitem construir *Description Schemes* estão apresentados no lado direito da figura 5.

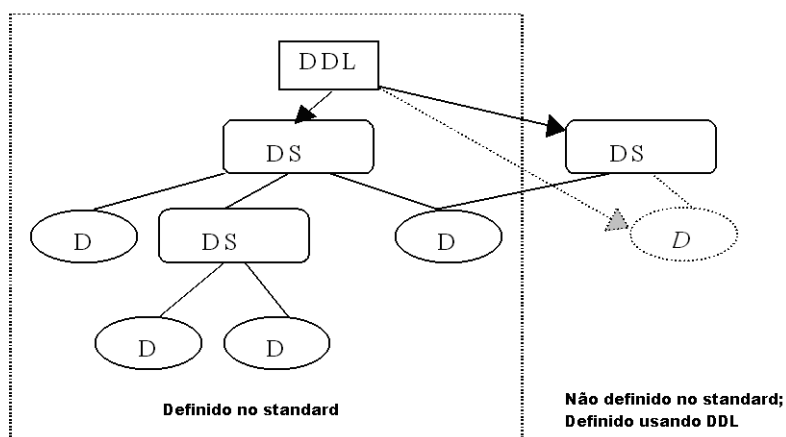


figura 5 - Representação das relações entre os elementos do standard *MPEG-7* [MARTINEZ, 2002]

4.2.2 Componentes *MPEG-7*

O *standard MPEG-7* contém vários componentes que permitem especificar descritores para vários tipos de conteúdos que são necessários indexar. Apesar de neste trabalho se abordar somente uma pequena parcela de todos os componentes que este *standard* disponibiliza, apresenta-se de seguida uma listagem com os vários componentes que fazem parte do mesmo.

O *standard MPEG-7* é composto pelos seguintes componentes [MARTINEZ, 2002]:

1. *Sistemas MPEG-7* – Ferramentas necessárias para realizar as descrições *MPEG-7*.
2. *MPEG-7 Description Definition Language* – A linguagem para definir novos DS's ou novos D's.
3. *MPEG-7 Audio* – Os D's e os DS's que dizem, respeito somente a descrições áudio.
4. *MPEG-7 Visual* - Os D's e os DS's que dizem, respeito somente a descrições vídeo.
5. *MPEG-7 Multimedia Description Schemes* - Os D's e os DS's que dizem respeito a descrições de multimédia.
6. *MPEG-7 Reference Software* – Uma implementação de *software* que diz respeito só às partes mais relevantes do *standard*.
7. *MPEG-7 Conformance* – Guias e procedimentos para testar a conformidade das implementações *MPEG-7*.

Destes componentes, o mais importante no desenvolvimento de aplicações que tenham como objectivo caracterizar informação multimédia, é o *MPEG-7 Multimedia Description Schemes (MDS)*.

4.2.3 *MDS - Multimedia Description Schemes*

O componente MDS tem como principal objectivo disponibilizar um conjunto de *Descriptor's* e *Description Schemes* que permitam descrever e lidar com entidades multimédia. É constituído por diferentes tipos de elementos que permitem caracterizar diversos aspectos relativos à descrição de conteúdos audiovisuais. Estes estão organizados em seis grupos distintos que são relativos aos vários aspectos que é necessário ter em consideração quando se está a construir um arquivo audiovisual.

4.2.3.1 Organização dos elementos do MDS:

Os elementos disponibilizados no *Multimedia Description Schemes* estão organizados para permitir a descrição das seguintes características [SALEMBIER, 2001]:

1. Descrição de elementos básicos
2. Descrição de conteúdos
3. Gestão de conteúdos
4. Organização de conteúdos
5. Navegação e acesso
6. Interacção com o utilizador

Através da utilização destes elementos é possível obter uma descrição bastante pormenorizada de todo o conteúdo multimédia que é necessário armazenar e caracterizar, durante a construção de um arquivo audiovisual.

A figura 6 mostra a organização dos elementos referidos previamente.

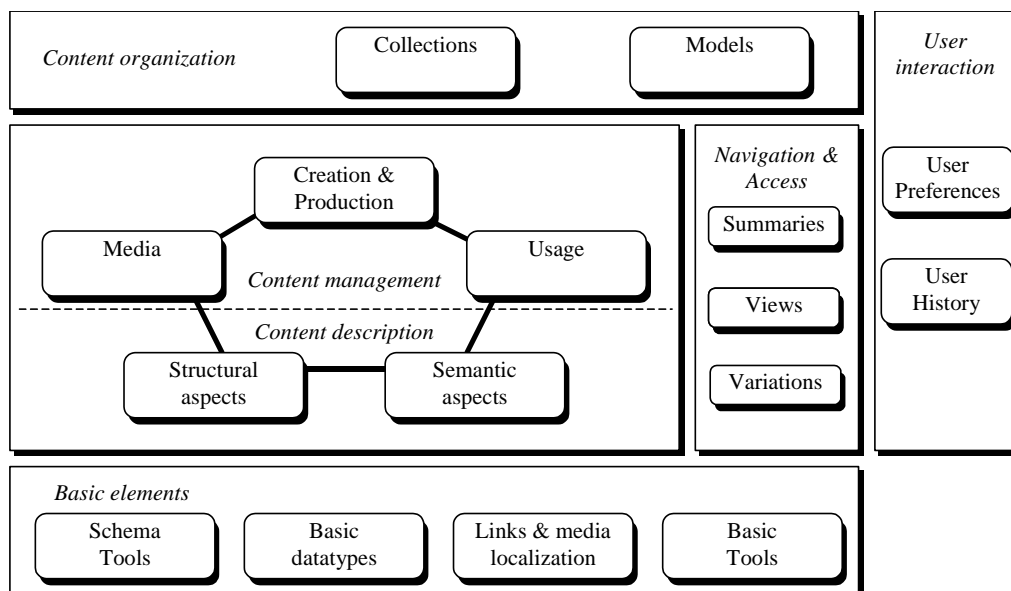


figura 6 - Organização dos elementos MDS [MARTINEZ, 2002]

4.2.3.2 Descrição dos elementos do *MDS*:

Apresentado o modo como se organizam os elementos do *MDS* descreve-se um pouco as funcionalidades a que estão associados.

Os elementos de organização do conteúdo permitem descrever a organização e a modelação de colecções de conteúdo audiovisual.

Os elementos de navegação e acesso disponibilizam funcionalidades que permitem recuperar conteúdo audiovisual através da definição de sumários, partições, decomposições e variações do conteúdo audiovisual.

Os elementos de interacção com o utilizador permitem descrever o histórico das preferências e utilização de conteúdo audiovisual.

Os elementos de descrição do conteúdo descrevem a estrutura (regiões, *frames* de vídeo, segmentos de áudio, etc.) e a semântica (objectos, eventos e noções abstractas) do conteúdo audiovisual.

Os elementos de gestão do conteúdo audiovisual servem para descrever aspectos relacionados com a criação, produção, codificação, armazenamento, formato dos ficheiros e sua utilização.

Os elementos básicos pertencentes ao *MDS* servem principalmente para descrever aspectos comuns a todo o conteúdo audiovisual como é o caso da descrição temporal, lugares, pessoas, indivíduos, grupos ou organizações.

4.2.4 Modelo de descrição

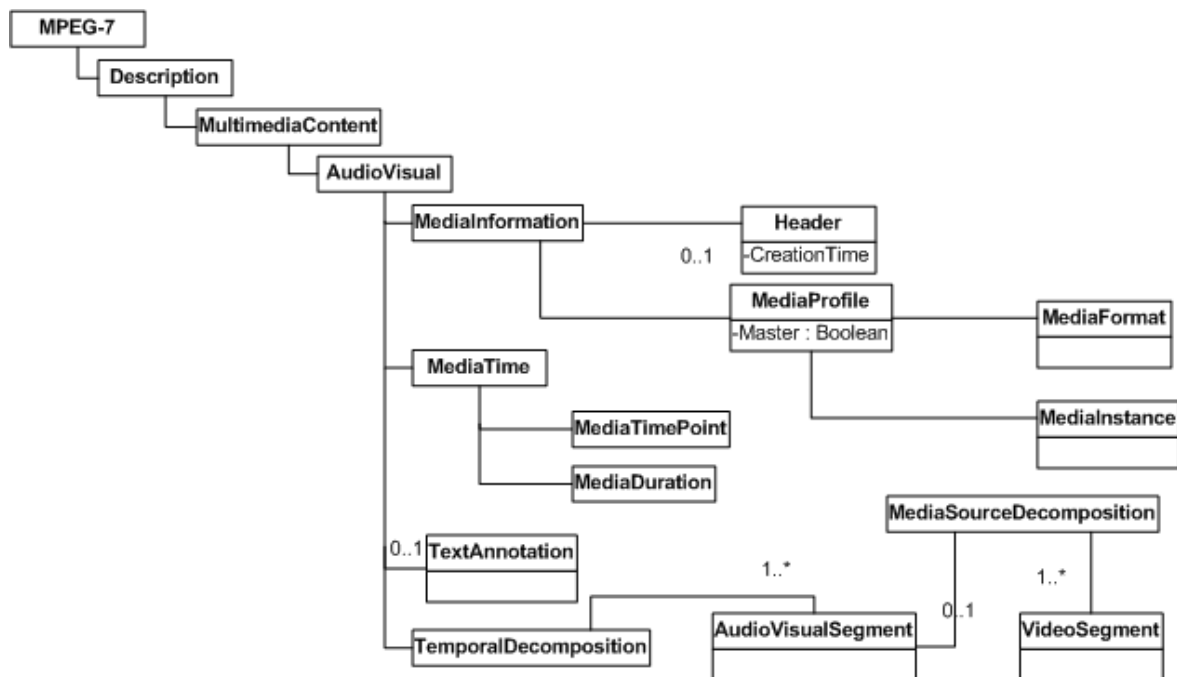
O *MPEG-7* dispõe de um conjunto de elementos que permitem caracterizar um conteúdo audiovisual em geral ou uma instância em particular.

Existem vários aspectos a analisar, tendo cada um deles uma correspondência com características de elementos pertencentes ao *MDS*.

Os aspectos mais relevantes e que é necessário caracterizar na construção de um arquivo audiovisual são os seguintes:

- *Conteúdo* → É um acontecimento em particular, que pode ser representado por vários tipos de média: áudio, vídeo ou audiovisual. Neste caso em particular descreve-se conteúdo audiovisual correspondente a um filme de uma sessão parlamentar.
- *Descrição física* → A descrição física de um conteúdo audiovisual é descrita com o *Media Information DS*.
- *Formatos* → Um conteúdo audiovisual pode estar representado em diferentes formatos. Pode existir um conteúdo principal (*master*) e diferentes perfis para várias representações do mesmo conteúdo.
- *Instâncias* → Um conteúdo audiovisual pode estar instanciado em uma entidade física concreta. Essa instância é caracterizada por uma identificação e uma localização.
- *Informação temporal* → Serve para caracterizar entidades temporais relativas a um conteúdo audiovisual. Descreve o seu instante inicial, a sua duração e outros aspectos relativos a informação temporal.
- *Partições e decomposições* → Descrevem diferentes decomposições do conteúdo audiovisual. Podem ser utilizadas para descrever diferentes vistas da informação audiovisual. Possibilita decompor um conteúdo audiovisual em vários segmentos. A descrição dos aspectos estruturais do conteúdo audiovisual é feita através do *Segment DS*. O seu objectivo é descrever os aspectos físicos e lógicos do conteúdo audiovisual. Em analogia com a programação por objectos, o *Segment DS* é uma classe abstracta e não pode ser instanciada directamente.

Utilizando elementos pertencentes ao *MDS* construiu-se um modelo para a descrição do conteúdo audiovisual. A figura 7 apresenta o modelo de descrição, criado com descritores *MPEG-7*, que permite descrever o conteúdo de um vídeo com as sessões parlamentares.

figura 7 - Modelo *MPEG-7* utilizado

Pode observar-se que o primeiro elemento que aparece é o elemento *MPEG-7*, indicando este que o conteúdo do ficheiro *XML* é uma descrição *MPEG-7*.

Seguidamente encontra-se o elemento *Description* que indica tratar-se de uma descrição.

O elemento *MultimediaContent* descreve o tipo de conteúdo a ser descrito.

O elemento *AudioVisual* representa o conteúdo audiovisual na sua totalidade. Contém os elementos *MediaInformation*, *MediaTime*, *TemporalDecomposition* e *TextAnnotation*.

O elemento *MediaInformation* contém informação relativa à codificação e localização do conteúdo. Contém o elemento *Header*, onde se armazena informação acerca da data de criação e o elemento *MediaProfile* que armazena informação sobre o formato e a instância que está ser caracterizada.

O elemento *MediaTime* contém informação relativa à duração do conteúdo audiovisual na totalidade. Contém o elemento *MediaTimePoint*, cujo conteúdo indica o instante inicial do filme e o elemento *MediaDuration* que indica a duração total do conteúdo audiovisual.

Os elementos *TextAnnotation* permitem colocar informação adicional para caracterizar o conteúdo multimédia, nomeadamente notas adicionais, anotações estruturadas e palavras chave.

O elemento *TemporalDecomposition*, como o próprio nome sugere, indica uma decomposição temporal do conteúdo multimédia. É composto por um ou mais elementos *AudioVisualSegment* que correspondem aos segmentos de vídeo do conteúdo audiovisual. Cada um destes segmentos vai corresponder à ocorrência de uma intervenção dos oradores no decorrer de uma Sessão Parlamentar.

Finalmente dentro dos elementos *AudioVisualSegment* temos os elementos *MediaSourceDecomposition* e *VideoSegment* que vão permitir caracterizar sub-segmentos de vídeo dentro de um segmento.

De seguida analisam-se pormenorizadamente os elementos descritos para se obter uma melhor imagem conceptual da utilização e funcionamento dos mesmos.

4.2.4.1 *MediaFormat*

Um descritor que convém destacar pela sua funcionalidade é o *MediaFormat*. Os elementos associados ao mesmo são de extrema utilidade quando se pretende construir um sistema de arquivo audiovisual.

Através do elemento *Content* pode-se identificar qual o tipo de informação armazenada: áudio, vídeo ou audiovisual.

Quando se pretende classificar em que tipo de suporte físico se encontra armazenado o conteúdo audiovisual utiliza-se o elemento *Medium*. Este elemento permite, por exemplo, indicar se a informação se encontra armazenada em *CD* ou no disco rígido de um computador.

O elemento *FileFormat* permite descrever em que formato de codificação se encontra o conteúdo. Pode-se, por exemplo, indicar se o formato de compressão do conteúdo audiovisual é *MPEG*, *Windows Media Video*, *Real Time*, etc.

O elemento *FileSize* permite indicar o tamanho do ficheiro em *bytes* que o conteúdo audiovisual ocupa.

O elemento *Bandwidth* e o elemento *BitRate* permitem indicar a largura de banda e o *BitRate* necessários para transmitir o conteúdo audiovisual.

O elemento *System* serve para indicar qual o sistema vídeo do conteúdo audiovisual, como por exemplo *PAL*.

Com o elemento *VisualCoding* podem-se indicar os parâmetros de codificação que foram utilizados. Contém o elemento *Format* onde se indica o formato do ficheiro, como por

exemplo *asf* ou *wmv*, e o elemento *Frame* onde se indica a *frame-rate* e a altura e largura do conteúdo audiovisual.

O elemento *AudioFormat* contém o elemento *Format* que permite caracterizar o formato de codificação áudio, o elemento *AudioChannels* que indica quantos canais são utilizados e o elemento *Sample* que permite caracterizar a *sample rate* do áudio.

Estes descritores permitem criar um sistema que descreva perfeitamente as condições em que se encontra armazenada a informação audiovisual. Permitem também saber em que tipo de suporte físico se encontra armazenada a informação. Este aspecto é importante pois num arquivo audiovisual a informação pode estar armazenada em *CD's* ou em Cassetes de Vídeo, mas o conteúdo dos mesmos estar descrito num ficheiro *MPEG-7* à parte.

No caso dos arquivos audiovisuais esta informação é bastante útil pois pode não ser possível ter todo o conteúdo audiovisual digitalizado e acessível através de um computador e muitas vezes o conteúdo audiovisual a ser caracterizado encontra-se em diferentes modalidades, formatos, codificações, etc. e podem mesmo existir diferentes instâncias do mesmo material.

A figura 8 apresenta o modo como o descritor *MediaFormat* se encontra organizado.

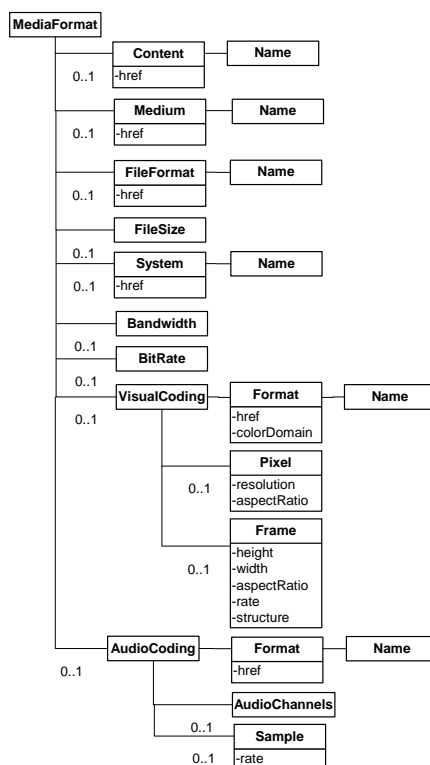


figura 8 - Descritor *MPEG-7 MediaFormat*

Apresenta-se em seguida na figura 9 o exemplo de uma descrição *XML* com um descritor *MediaFormat*.

```
<MediaFormat>
  <Content href="MPEG7ContentCS">
    <Name>audiovisual</Name>
  </Content>
  <Medium href="MPEG7MediumCS">
    <Name>Hard Disk</Name>
  </Medium>
  <FileFormat href="MPEG7FileFormatCS">
    <Name>Windows Media Video</Name>
  </FileFormat>
  <FileSize>200000000</FileSize>
  <System href="MPEG7SystemCS">
    <Name>PAL</Name>
  </System>
  <Bandwidth>1.5e6</Bandwidth>
  <BitRate>1500000</BitRate>
  <VisualCoding>
    <Format href="MPEG7FileFormatCS" colorDomain="color">
      <Name>asf</Name>
    </Format>
    <Pixel resolution="2" aspectRatio="1"/>
      <Frame height="240" width="320" aspectRatio="1.0"
        rate="30.0" structure="progressive"/>
    </Pixel>
  </VisualCoding>
  <AudioCoding>
    <Format href="MPEG7FileFormatCS">
      <Name>Windows Media Audio</Name>
    </Format>
    <AudioChannels>2</AudioChannels>
    <Sample rate="44100"/>
  </AudioCoding>
</MediaFormat>
```

figura 9 - Descrição *XML* com o descritor *MediaFormat*

4.2.4.2 *MediaInstance*

O descritor *MediaInstance* tem como principal funcionalidade descrever qual o *URI* (*Universal Resource Identifier*) onde se encontra o conteúdo audiovisual que está a ser descrito.

O *URI* pode indicar quer a localização do vídeo num conjunto de servidores, quer a localização do vídeo dentro de um CD ou outro suporte físico.

O elemento *InstanceIdentifier* permite armazenar a identificação do conteúdo audiovisual.

O elemento *MediaLocator* contém o elemento *MediaUri* que armazena o *URI* com a localização do conteúdo audiovisual.

A figura 10 apresenta o modo como se encontram organizados os elementos que este descritor possui.

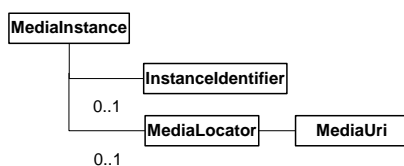


figura 10 - Descritor *MPEG-7 MediaInstance*

A figura 11 apresenta um exemplo de uma descrição *XML* com o descritor *MediaInstance*.

```

<MediaInstance>
  <InstanceIdentifier>S1L8SL1N2</InstanceIdentifier>
  <MediaLocator>
    <MediaUri>http://tribunal.ieeta.pt/S1L8SL1N2.wmv</MediaUri>
  </MediaLocator>
</MediaInstance>
  
```

figura 11 - Descrição *XML* com o descritor *MediaInstance*

4.2.4.3 *AudioVisualSegment*

Após descrever a informação relativa ao formato e localização do conteúdo audiovisual surge o momento em que se devem caracterizar os vários segmentos em que o conteúdo audiovisual se decompõe.

O descritor *AudioVisualSegment* serve para decompor um conteúdo audiovisual em pequenas parcelas. Este descritor contém o descritor *TextAnnotation*, *MediaTime* e *MediaSourceDecomposition*.

Com o descritor *TextAnnotation* pode-se registar informação adicional sobre o segmento de vídeo. Na secção 4.2.4.5 indica-se o tipo de informação que se pode inserir com este descritor.

O descritor *MediaTime* contém o elemento *MediaRelTimePoint* que permite registar o início do segmento de vídeo e o elemento *MediaDuration* que permite registar a duração do mesmo.

O descritor *MediaSourceDecomposition* tem como função dividir segmentos de vídeo em sub-segmentos. Deste modo, pode obter-se uma alta granularidade na indexação do vídeo. Este descritor por sua vez pode conter um ou mais descritores *VideoSegment*.

A figura 12 apresenta o modo como os elementos pertencentes ao descritor *AudioVisualSegment* se encontram organizados.

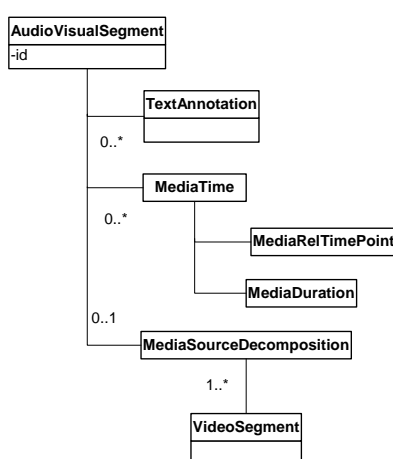


figura 12 - Descritor *MPEG-7 AudioVisualSegment*

A figura 13 apresenta um exemplo de uma descrição *XML* com o descritor *AudioVisualSegment*.

```
<AudioVisualSegment id="3">
  <MediaTime>
    <MediaRelTimePoint>PT02H07M19S</MediaRelTimePoint>
    <MediaDuration>PT00H05M29S</MediaDuration>
  </MediaTime>
  <MediaSourceDecomposition>
    <VideoSegment id="Sub-intervencao1">
      <TextAnnotation>
        <FreeTextAnnotation>Intervenção Parlamentar</FreeTextAnnotation>
      </TextAnnotation>
      <MediaTime>
        <MediaTimePoint>T02:07:50</MediaTimePoint>
        <MediaDuration>PT1M00S</MediaDuration>
      </MediaTime>
    </VideoSegment>
    <VideoSegment id="Sub-intervencao2">
      <MediaTime>
        <MediaTimePoint>T02:09:11</MediaTimePoint>
        <MediaDuration>PT1M30S</MediaDuration>
      </MediaTime>
    </VideoSegment>
  </MediaSourceDecomposition>
</AudioVisualSegment>
```

figura 13 - Descrição *XML* com o descritor *AudioVisualSegment*

4.2.4.4 *VideoSegment*

O descritor *VideoSegment* existe dentro do descritor *MediaSourceDecomposition*. Este descritor tem como funcionalidade caracterizar sub-segmentos de vídeo dentro de um segmento. Pode conter um descritor *TextAnnotation* para adicionar informação textual ao sub-segmento de vídeo. Contém também o descritor *MediaTime* que permite caracterizar o sub-segmento de vídeo quanto à informação temporal.

Na figura 14 podemos observar o modo como é composto descritor *VideoSegment*.

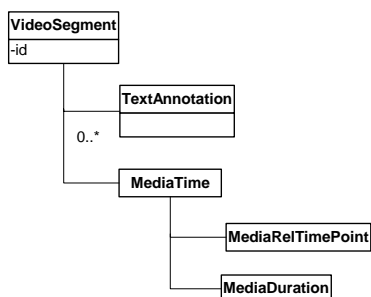


figura 14 - Descritor *MPEG-7 VideoSegment*

A figura 15 apresenta um exemplo de uma descrição *XML* com o descritor *VideoSegment*.

```

<VideoSegment id="Sub-intervencao1">
  <TextAnnotation>
    <FreeTextAnnotation>Notas de sub intervenção</FreeTextAnnotation>
  </TextAnnotation>
  <MediaTime>
    <MediaTimePoint>T02:07:50</MediaTimePoint>
    <MediaDuration>PT1M00S</MediaDuration>
  </MediaTime>
</VideoSegment>

```

figura 15 - Descrição *XML* com o descritor *VideoSegment*

4.2.4.5 *TextAnnotation*

O descritor *TextAnnotation* permite inserir notas de texto associadas quer ao conteúdo em geral quer a um segmento em particular. Este descritor contém três elementos que servem para criar anotações de texto livre, anotações estruturadas e inserir palavras-chave.

O elemento *FreeTextAnnotation* permite introduzir informação descritiva de uma forma livre, isto é, texto corrente.

O elemento *StructuredAnnotation* permite fazer uma descrição de informação de um modo estruturado. Esta estrutura pode conter informação sobre quem, qual objecto, que acção,

onde, quando, porquê e como. Este tipo de anotação permite aumentar o nível de detalhe na descrição. Deste modo, podem construir-se sistemas de pesquisa mais elaborados que permitam realizar pesquisas mais rebuscadas dentro do conteúdo audiovisual, uma vez que existem vários campos que podem ser pesquisados de um modo *booleano*.

Finalmente para adicionar descritores ao conteúdo audiovisual utiliza-se o elemento *KeywordAnnotation*. Este permite associar uma lista de descritores a um determinado conteúdo audiovisual.

Todos estes elementos referidos anteriormente permitem associar descrições textuais ao conteúdo audiovisual. Deste modo é possível construir sistemas de pesquisa que permitam obter segmentos de vídeo com conteúdos temáticos específicos.

A figura 16 mostra o modo como o descritor *TextAnnotation* está estruturado.

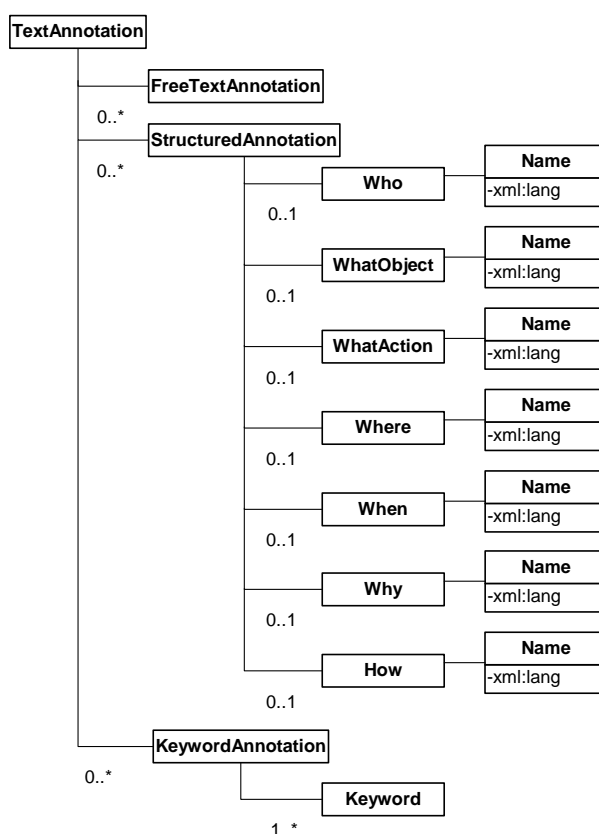


figura 16 - Descritor *MPEG-7 TextAnnotation*

A figura 17 apresenta um exemplo de uma descrição *XML* com o descritor *TextAnnotation* e os elementos que fazem parte do mesmo.

```
<TextAnnotation>
  <FreeTextAnnotation>Notas sobre a intervenção</FreeTextAnnotation>
  <KeywordAnnotation>
    <Keyword>pesca</Keyword>
    <Keyword>agricultura</Keyword>
    <Keyword>comércio</Keyword>
  </KeywordAnnotation>
  <StructuredAnnotation>
    <Who>
      <Name>Informação sobre o sujeito</Name>
    </Who>
    <Where>
      <Name>Descrição sobre o local</Name>
    </Where>
    <When>
      <Name>Descrição temporal</Name>
    </When>
  </StructuredAnnotation>
</TextAnnotation>
```

figura 17 - Descrição *XML* com o descritor *TextAnnotation*

4.2.5 Exemplo de uma descrição *MPEG-7*

Depois de criado o modelo *MPEG-7* de descrição do conteúdo audiovisual que se pretende caracterizar, apresenta-se na figura 18 um exemplo de uma possível descrição de um conteúdo audiovisual, de acordo com o modelo desenvolvido.

O elemento *MPEG7* é o primeiro elemento que aparece dentro do documento *XML* e contém a indicação do *XML-Schema* que permite validar o mesmo.

Seguidamente aparece o elemento *Description* com o atributo *ContentEntityType* para o documento ser válido de acordo com o *XML-Schema* definido.

O elemento *MultimediaContent* indica o tipo de conteúdo que está a ser analisado.

O elemento *MediaInformation* contém elementos que permitem caracterizar o tempo de criação, e o perfil do conteúdo audiovisual.

O elemento *Header* contém o elemento *CreationTime* que por sua vez contém informação relativa à data de criação.

O elemento *MediaProfile* com o atributo *master* indica se este é o conteúdo original ou se é uma cópia. Contém os elementos *MediaFormat* e *MediaInstance*.

Dentro do elemento *MediaFormat* existem os vários elementos que permitem caracterizar o formato do conteúdo analisado. Neste caso em particular pode dizer-se que quanto o tipo de conteúdo é audiovisual, que o suporte físico em que se encontra armazenado o mesmo é em disco rígido, que o formato do ficheiro é *Windows Media Video*, que o ficheiro que contém o vídeo tem uma tamanho de 200000000 *bytes*, que o sistema de vídeo é *PAL*, que a largura de banda de vídeo é 1.5e6 *bps* e que o *BitRate* do vídeo é de 150000 *bits/s*.

Quanto à codificação visual o formato de vídeo é *wmv*, os píxeis tem uma resolução de 2 e um *aspectRatio* de 1, as *frames* de vídeo tem uma estrutura progressiva com uma taxa de 30 imagens por segundo, uma altura de 240 píxeis e uma largura de 320 píxeis.

Quanto à codificação de áudio o formato do áudio é *Windows Media Audio*, com dois canais de áudio e uma taxa de amostragem de 44100.

O elemento *MediaInstance* contém o elemento *InstanceIdentifier* indicando que a identificação do vídeo é S1L8SL1N2 e o elemento *MediaLocator* com o elemento *MediaUri* indicando que o vídeo está no endereço <http://tribunal.ieeta.pt/s1l8sl1n2.wmv>.

No elemento *TextAnnotation* colocou-se uma nota indicando que o conteúdo apresenta uma sessão parlamentar.

Com os elementos *MediaTime*, *MediaRelTimePoint* e *MediaDuration* indica-se que o filme tem uma duração de 3 horas, 1 minuto e trinta segundos.

O elemento *TemporalDecomposition* contém três elementos *AudioVisualSegment* que indicam que este vídeo contém três segmentos de vídeo.

O primeiro segmento de vídeo, que corresponde ao elemento *AudioVisualSegment* com o atributo *id* igual a 1, representa um caso simples de um segmento de vídeo que têm início no instante inicial do vídeo e tem uma duração de 48 segundos, ao qual apenas se adicionou uma pequena nota textual sobre o conteúdo do segmento de vídeo.

O segundo segmento de vídeo, que corresponde ao elemento *AudioVisualSegment* com o atributo *id* igual a 2, representa um caso de um segmento de vídeo com início às 2 horas e 6 minutos e com uma duração de 32 segundos.

O terceiro segmento de vídeo, que corresponde ao elemento *AudioVisualSegment* com o atributo *id* igual a 3, representa um caso de um segmento de vídeo que é decomposto em sub-segmentos. O instante inicial do segmento de vídeo é às 2 horas, 7 minutos e 19 segundos e tem uma duração de 5 minutos e 29 segundos. Este segmento por sua vez contém dois sub-segmentos de vídeo. O primeiro sub-segmento tem início às 2 horas, 7 minutos e 50 segundos e uma duração de 1 minuto. Contém também uma pequena nota textual a indicar que se trata de uma sub-intervenção. O segundo sub-segmento de vídeo teve início no instante 2 horas, 9 minutos e 11 segundos e tem uma duração de 1 minuto e 30 segundos.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 .\Mpeg7-2001.xsd">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="AudioVisualType">
      <AudioVisual>
        <MediaInformation>
          <Header xsi:type="DescriptionMetadataType">
            <CreationTime>2001-01-08</CreationTime>
          </Header>
          <MediaProfile master="true">
            <MediaFormat>
              <Content href="MPEG7ContentCS">
                <Name>audiovisual</Name>
              </Content>
              <Medium href="MPEG7MediumCS">
                <Name>Hard Disk</Name>
              </Medium>
              <FileFormat href="MPEG7FileFormatCS">
                <Name>Windows Media Video</Name>
              </FileFormat>
              <FileSize>200000000</FileSize>
              <System href="MPEG7SystemCS">
                <Name>PAL</Name>
              </System>
              <Bandwidth>1.5e6</Bandwidth>
              <BitRate>150000</BitRate>
              <VisualCoding>
                <Format href="MPEG7FileFormatCS" colorDomain="color">
                  <Name>asf</Name>
                </Format>
```

```
<Pixel resolution="2" aspectRatio="1"/>
  <Frame height="240" width="320" aspectRatio="1.0" rate="30.0"
structure="progressive"/>
</VisualCoding>
<AudioCoding>
  <Format href="MPEG7FileFormatCS">
    <Name>Windows Media Audio</Name>
  </Format>
  <AudioChannels>2</AudioChannels>
  <Sample rate="44100"/>
</AudioCoding>
</MediaFormat>
<MediaInstance>
  <InstanceIdentifier>S1L8SL1N2</InstanceIdentifier>
  <MediaLocator>
    <MediaUri>http://tribunal.ieeta.pt/S1L8SL1N2.wmv</MediaUri>
  </MediaLocator>
</MediaInstance>
</MediaProfile>
</MediaInformation>
<TextAnnotation>
  <FreeTextAnnotation> Sessão Parlamentar </FreeTextAnnotation>
</TextAnnotation>
<MediaTime>
  <MediaTimePoint>T00:00:00</MediaTimePoint>
  <MediaDuration>PT3H1M30S</MediaDuration>
</MediaTime>
<TemporalDecomposition>
  <AudioVisualSegment id="1">
    <TextAnnotation>
      <FreeTextAnnotation>Intervenção Parlamentar</FreeTextAnnotation>
    </TextAnnotation>
    <MediaTime>
      <MediaRelTimePoint>PT00H00M00S</MediaRelTimePoint>
      <MediaDuration>PT00H00M48S</MediaDuration>
    </MediaTime>
  </AudioVisualSegment>
  <AudioVisualSegment id="2">
    <TextAnnotation>
      <FreeTextAnnotation>Notas sobre a intervenção</FreeTextAnnotation>
      <KeywordAnnotation>
        <Keyword>pesca</Keyword>
        <Keyword>agricultura</Keyword>
        <Keyword>comércio</Keyword>
      </KeywordAnnotation>
```

```

<StructuredAnnotation>
  <Who>
    <Name>Informação sobre o sujeito</Name>
  </Who>
  <Where>
    <Name>descrição sobre o local</Name>
  </Where>
  <When>
    <Name>descrição temporal</Name>
  </When>
</StructuredAnnotation>
</TextAnnotation>
<MediaTime>
  <MediaRelTimePoint>PT02H06M00S</MediaRelTimePoint>
  <MediaDuration>PT00H00M32S</MediaDuration>
</MediaTime>
</AudioVisualSegment>
<AudioVisualSegment id="3">
  <MediaTime>
    <MediaRelTimePoint>PT02H07M19S</MediaRelTimePoint>
    <MediaDuration>PT00H05M29S</MediaDuration>
  </MediaTime>
  <MediaSourceDecomposition>
    <VideoSegment id="Sub-intervencao1">
      <TextAnnotation>
        <FreeTextAnnotation>Notas de sub intervenção</FreeTextAnnotation>
      </TextAnnotation>
      <MediaTime>
        <MediaTimePoint>T02:07:50</MediaTimePoint>
        <MediaDuration>PT1M00S</MediaDuration>
      </MediaTime>
    </VideoSegment>
    <VideoSegment id="Sub-intervencao2">
      <MediaTime>
        <MediaTimePoint>T02:09:11</MediaTimePoint>
        <MediaDuration>PT1M30S</MediaDuration>
      </MediaTime>
    </VideoSegment>
  </MediaSourceDecomposition>
</AudioVisualSegment>
</TemporalDecomposition>
</AudioVisual>
</MultimediaContent>
</Description>
</Mpeg7>

```

figura 18 - Exemplo de uma descrição de um conteúdo audiovisual

4.3 DATA BINDING

4.3.1 Introdução

Após obter o modelo para descrever os vídeos é necessário construir uma aplicação que permita gerar e manipular os documentos que contêm as descrições dos vídeos. Este sub-capítulo apresenta o conceito de *Data Binding*, que tem por objectivo transformar documentos *XML* em objectos que posteriormente podem ser manipulados numa aplicação.

4.3.2 Arquitectura JAXB (*Java Architecture for XML Binding*)

Na programação por objectos existe um conceito fundamental que é a distinção entre classes e objectos. Uma classe tem como característica modelar enquanto que um objecto é uma instância de uma classe em particular.

Existe uma analogia semelhante no *XML*. Normalmente utiliza-se um *XML Schema* ou um *DTD* para descrever a estrutura permitida de um determinado documento *XML*. Após esta descrição torna-se possível validar um documento *XML* de acordo com o *XML Schema* ou *DTD* definido.

A ideia que está por detrás do *Data Binding* é bastante similar e tem por objectivo encontrar uma forma de criar uma correspondência entre a descrição de uma estrutura de um documento *XML* e uma classe.

A linguagem de programação orientada a objectos *JAVA* tem um pacote de ferramentas, que tem por nome *JAXB (Java Architecture for XML Binding)* [SUN, 2003a], que permite implementar estes conceitos.

A funcionalidade que está associada ao *JAXB* é a possibilidade de mapear documentos *XML* em objectos de *JAVA*.

O *JAXB* disponibiliza um conjunto de *API's* que permitem gerar classes *JAVA* a partir de *XML Schemas*, utilizando um compilador de *Schemas*, e converter ficheiros *XML* para objectos *JAVA*. [SUN, 2003b]

A figura 19 representa a arquitectura da tecnologia *JAXB*. Para uma melhor compreensão dos componentes que compõem a mesma e o seu modo de funcionamento descrevem-se em seguida as funcionalidades a eles associadas.

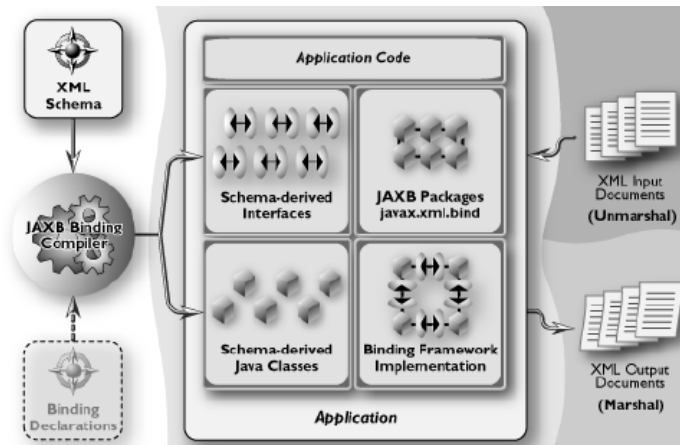


figura 19 - Arquitectura da ferramenta *JAXB* [SUN, 2003b]

O *XML Schema* serve como ponto de partida para definir uma classe de documentos *XML*. É a partir deste *Schema* que são geradas as classes *Java* para representação da informação. As *Binding Declarations* servem para definir regras adicionais que não estejam claramente definidas no *XML Schema*. Servem como extensão às funcionalidades inerentes aos *XML Schemas*.

O *Binding Compiler* é a parte fulcral do processo de *Data Binding* do *JAXB*. É a partir deste compilador que são geradas as classes de *Java* que respeitam as regras que estão definidas no *XML Schema*.

O *Binding Framework Implementation* é uma *API* que disponibiliza os *interfaces* necessários para os processos de *unmarshalling*, *marshalling* e validação do conteúdo *XML* dentro uma aplicação *JAVA*.

Schema Derived Classes são as classes geradas pelo *JAXB Binding Compiler*. Estas classes são específicas para cada *XML Schema*.

JAVA Application é uma aplicação cliente *JAVA* que utiliza todos estes conceitos ligados ao *framework JAXB*, como por exemplo o *marshalling*, *unmarshalling* e validação de informação representada em *XML*.

XML input documents são os documentos *XML* que servem como entrada para o *framework JAXB*. Não tem de ser necessariamente um documento *XML* bem formado nem validado.

XML output document é um conteúdo *XML* que é *marshalled* para um documento *XML*. No *JAXB* o *marshalling* envolve o *parsing* de uma árvore com conteúdo *XML* para um documento *XML* que é uma representação exacta de um documento original *XML* e é válido de acordo com o *XML Schema* que serviu de modelo. O *JAXB* pode fazer o *marshall* dos dados *XML* para documentos *XML*, para *SAX content handlers* e nodos *DOM*. O *unmarshalling* envolve o *parsing* de um conteúdo *XML* para um objecto *JAVA*.

4.3.3 Processo de *Data Binding* no *JAXB*.

O processo de *Data Binding* é efectuado seguindo um conjunto de passos :

- 1.º Gerar Classes – É utilizado um *XML Schema* como entrada para o compilador de *JAXB binding* para gerar classes *JAXB* baseadas no *Schema* utilizado.
- 2.º Compilar Classes – Todas as classes geradas, ficheiros fonte e código têm de ser compilados.
- 3.º *Unmarshal* – A partir de documentos *XML* válidos de acordo com o *XML Schema* fonte, pode-se efectuar o *unmarshall* através do *framework JAXB*. O *JAXB* também suporta *unmarshalling* de informação de outras fontes.
- 4.º Gerar árvore de conteúdos – O processo de *unmarshalling* gera uma árvore de conteúdo de objectos instanciados a partir de classes *JAXB*. Esta árvore de conteúdos representa a estrutura e conteúdo dos documentos *XML* de entrada.
- 5.º Validação (opcional) – O processo de *unmarshalling* envolve opcionalmente a validação de documentos *XML* de entrada antes da geração da árvore de conteúdos. Também se pode utilizar a validação antes do processo de *marshalling* de conteúdo *XML* para um documento *XML*.
- 6.º Processamento de conteúdos – A aplicação cliente pode modificar a informação *XML* representada sob a forma de uma árvore de conteúdo *JAVA* através de interfaces gerados pelo *JAXB binding compiler*.

7.º Marshall – O conteúdo da árvore processado é *marshalled* para um ou mais documentos *XML* de saída. O conteúdo pode ser validado antes do processo de *marshalling*.

A figura 20 ilustra os passos gerais utilizados no processo de *DATA Binding* numa aplicação em que se utiliza o *framework* do *JAXB*.

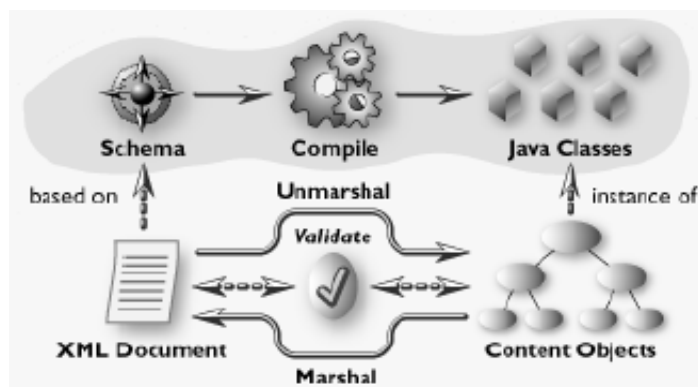


figura 20 - Passos gerais do processo de *Data Binding* [SUN, 2003b]

Com este *framework* do *JAXB* pode-se, a partir de um *XML Schema* base, gerar objectos com informação *XML*, alterar informação, criar novos objectos e por fim é possível voltar a gravar os objectos sob a forma de documentos *XML*, validados de acordo o *XML Schema* utilizado.

4.3.4 Descrição da estrutura do *XML Schema*

Para iniciar o processo é necessário obter um *XML Schema* que descreva os documentos *XML* para que o *JAXB Binding Compiler* possa gerar as classes Java que permitem manipular os documentos *XML*.

Um *XML Schema* é ele próprio um ficheiro *XML* daí que as duas primeiras linhas do *XML Schema* sejam as seguintes:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
```

O primeiro elemento que aparece nos documentos XML com as descrições dos vídeo é o elemento *MPEG7*. Este elemento é composto por uma sequência de elementos de *Description*, que estão associados ao tipo *DescriptionType*.

```
<xs:element name="Mpeg7">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Description" type="DescriptionType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Os elementos do tipo *DescriptionType* contêm elementos do tipo *MultimediaContentType*. Estes dois tipos permitem criar elementos *Description* e *MultimediaContent* que são definidos no *XML Schema* do seguinte modo:

```
<xs:complexType name="DescriptionType">
  <xs:sequence>
    <xs:element name="MultimediaContent" type="MultimediaContentType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="MultimediaContentType">
  <xs:sequence>
    <xs:element name="AudioVisual" type="AudioVisualType"/>
  </xs:sequence>
</xs:complexType>
```

O elemento *AudioVisual* é composto por uma sequência de elementos do tipo *MediaInformation*, *TextAnnotation*, *MediaTime* e *TemporalDecomposition*.

```
<xs:complexType name="AudioVisualType">
  <xs:sequence>
    <xs:element name="MediaInformation" type="MediaInformationType"/>
    <xs:element name="TextAnnotation" type="TextAnnotationType"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="MediaTime" type="MediaTimeType"/>
    <xs:element name="TemporalDecomposition"
type="TemporalDecompositionType"/>
  </xs:sequence>
</xs:complexType>
```

O elemento *MediaInformation* é composto por uma sequência de elementos *Header* e *MediaProfile*.

```
<xs:complexType name="MediaInformationType">
  <xs:sequence>
    <xs:element name="Header" type="HeaderType"/>
    <xs:element name="MediaProfile" type="MediaProfileType"/>
  </xs:sequence>
</xs:complexType>
```

O elemento *Header* contém o elemento *CreationTime* onde se armazena o valor da data de criação.

```
<xs:complexType name="HeaderType">
  <xs:sequence>
    <xs:element ref="CreationTime"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="CreationTime" type="xs:string"/>
```

Os elementos do tipo *MediaProfileType* são compostos por uma sequência de elementos *MediaFormat* e *MediaInstance*. Possuem também definido um atributo opcional , *master*, do tipo booleano.

```
<xs:complexType name="MediaProfileType">
  <xs:sequence>
    <xs:element name="MediaFormat" type="MediaFormatType"/>
    <xs:element name="MediaInstance" type="MediaInstanceType"/>
  </xs:sequence>
  <xs:attribute name="master" type="xs:boolean" use="optional"/>
</xs:complexType>
```

Os elementos *MediaFormat* são compostos por uma sequência de elementos *Content*, *Medium*, *FileFormat*, *FileSize*, *System*, *Bandwidth*, *BitRate*, *VisualCoding* e *AudioCoding* que permitem definir os aspectos relacionados com o formato e codificação do conteúdo audiovisual. Por sua vez alguns destes elementos são complexos o que leva que ainda existam elementos que aparecem definidos dentro dos mesmos.

```
<xs:complexType name="MediaFormatType">
  <xs:sequence>
    <xs:element name="Content" type="ContentType"/>
    <xs:element name="Medium" type="MediumType" minOccurs="0"/>
    <xs:element name="FileFormat" type="FileFormatType"/>
    <xs:element ref="FileSize"/>
    <xs:element name="System" type="SystemType" minOccurs="0"/>
    <xs:element ref="Bandwidth" minOccurs="0"/>
    <xs:element ref="BitRate" minOccurs="0"/>
    <xs:element name="VisualCoding" type="VisualCodingType"/>
    <xs:element name="AudioCoding" type="AudioCodingType"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ContentType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="NameType">
  <xs:simpleContent>
    <xs:extension base="xs:string"/>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="MediumType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FileFormatType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="FileSize" type="xs:int"/>

<xs:complexType name="SystemType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>
```

```
<xs:element name="Bandwidth" type="xs:double"/>

<xs:element name="BitRate" type="xs:int"/>

<xs:complexType name="VisualCodingType">
  <xs:sequence>
    <xs:element name="Format" type="FormatType"/>
    <xs:element name="Pixel" type="PixelType" minOccurs="0"/>
    <xs:element name="Frame" type="FrameType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FormatType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
  <xs:attribute name="href" type="xs:string" use="optional"/>
  <xs:attribute name="colorDomain" type="xs:string"/>
</xs:complexType>

<xs:complexType name="PixelType">
  <xs:attribute name="resolution" type="xs:byte" use="required"/>
  <xs:attribute name="aspectRatio" type="xs:boolean" use="required"/>
</xs:complexType>

<xs:complexType name="FrameType">
  <xs:attribute name="height" type="xs:short" use="required"/>
  <xs:attribute name="width" type="xs:short" use="required"/>
  <xs:attribute name="aspectRatio" type="xs:decimal" use="optional"/>
  <xs:attribute name="rate" type="xs:decimal" use="optional"/>
  <xs:attribute name="structure" type="xs:string" use="optional"/>
</xs:complexType>

<xs:complexType name="AudioCodingType">
  <xs:sequence>
    <xs:element name="Format" type="FormatType"/>
    <xs:element ref="AudioChannels" minOccurs="0"/>
    <xs:element name="Sample" type="SampleType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="AudioChannels" type="xs:int"/>

<xs:complexType name="SampleType">
  <xs:attribute name="rate" type="xs:int" use="required"/>
</xs:complexType>
```

Os elementos do tipo *MediaInstanceType* contêm o elemento *InstanceIdentifier* que é onde se armazena o valor de identificação do conteúdo audiovisual. Possuem também definido o elemento *MediaLocator* que permite armazenar o URL de localização.

```
<xs:complexType name="MediaInstanceType">
  <xs:sequence>
    <xs:element ref="InstanceIdentifier"/>
    <xs:element name="MediaLocator" type="MediaLocatorType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="InstanceIdentifier" type="xs:string"/>
<xs:complexType name="MediaLocatorType">
  <xs:sequence>
    <xs:element ref="MediaUri"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="MediaUri" type="xs:anyURI"/>
```

Os elementos do tipo *TextAnnotationType* contêm as anotações textuais. Podem conter elementos *FreeTextAnnotation*, *KeywordAnnotation* e *StructuredAnnotation*. Encontram-se definidos em seguida.

```
<xs:complexType name="TextAnnotationType">
  <xs:sequence>
    <xs:element ref="FreeTextAnnotation" minOccurs="0"/>
    <xs:element name="KeywordAnnotation" type="KeywordAnnotationType"
minOccurs="0"/>
    <xs:element name="StructuredAnnotation"
type="StructuredAnnotationType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="FreeTextAnnotation" type="xs:string"/>

<xs:complexType name="KeywordAnnotationType">
  <xs:sequence>
    <xs:element ref="Keyword" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="Keyword" type="xs:string"/>
```

```
<xs:complexType name="StructuredAnnotationType">
  <xs:sequence>
    <xs:element name="Who" minOccurs="0" type="WhoType"/>
    <xs:element name="WhatObject" minOccurs="0" type="WhatObjectType"/>
    <xs:element name="WhatAction" minOccurs="0" type="WhatActionType"/>
    <xs:element name="Where" minOccurs="0" type="WhereType"/>
    <xs:element name="When" minOccurs="0" type="WhenType"/>
    <xs:element name="Why" minOccurs="0" type="WhyType"/>
    <xs:element name="How" minOccurs="0" type="HowType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WhoType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WhatObjectType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WhatActionType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WhereType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WhenType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="WhyType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="HowType">
  <xs:sequence>
    <xs:element name="Name" type="NameType"/>
  </xs:sequence>
</xs:complexType>
```


Os elementos *MediaTime* são do tipo *MediaTimeType*. Podem conter o elemento *MediaTimePoint*, *MediaRelTimePoint* e o elemento *MediaDuration*. Estes estão definidos conforme se apresenta de seguida:

```
<xs:complexType name="MediaTimeType">
  <xs:sequence>
    <xs:element ref="MediaTimePoint" minOccurs="0"/>
    <xs:element ref="MediaRelTimePoint" minOccurs="0"/>
    <xs:element ref="MediaDuration"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="MediaTimePoint" type="mediaTimePointType"/>
<xs:simpleType name="mediaTimePointType">
  <xs:restriction base="xs:string">
    <xs:pattern value="(\-?\d+(\-\d{2})(\-\d{2})?)?(T\d{2}(:\d{2}(:\d{2}(:\d+)?))?)?(F\d+)?"/>
  </xs:restriction>
</xs:simpleType>

<xs:element name="MediaRelTimePoint" type="mediaDurationType"/>
<xs:simpleType name="mediaDurationType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\-?P(\d+D)?(T(\d+H)?(\d+M)?(\d+S)?(\d+N)?)(\d+F)?"/>
  </xs:restriction>
</xs:simpleType>

<xs:element name="MediaDuration" type="mediaDurationType"/>
```

O elemento *TemporalDecomposition* contém uma sequência de elementos *AudioVisualSegment*. Encontra-se definido do seguinte modo:

```
<xs:complexType name="TemporalDecompositionType">
  <xs:sequence>
    <xs:element name="AudioVisualSegment" type="AudioVisualSegmentType"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Os elementos *AudioVisualSegment* contém obrigatoriamente um elemento *MediaTime* e podem ter elementos *TextAnnotation* ou elementos *MediaSourceDecomposition*.

```
<xs:complexType name="AudioVisualSegmentType">
  <xs:sequence>
    <xs:element name="TextAnnotation" type="TextAnnotationType"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="MediaTime" type="MediaTimeType"/>
    <xs:element name="MediaSourceDecomposition" type="
MediaSourceDecompositionType" minOccurs="0" />
  </xs:sequence>
  <xs:attribute name="id" use="required"/>
</xs:complexType>
```

Os elementos *MediaSourceDecomposition* contém uma sequência de elementos *VideoSegment*.

```
<xs:complexType name="MediaSourceDecompositionType">
  <xs:sequence>
    <xs:element name="VideoSegment" type="VideoSegmentType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Finalmente os elementos *VideoSegment* contém obrigatoriamente um elemento *MediaTime* e podem conter elementos *TextAnnotation*.

Por último apresenta-se o fim de *XML Schema* porque este elemento foi o primeiro elemento definido.

```
<xs:complexType name="VideoSegmentType">
  <xs:sequence>
    <xs:element name="TextAnnotation" type="TextAnnotationType"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="MediaTime" type="MediaTimeType"/>
  </xs:sequence>
  <xs:attribute name="id" use="required"/>
</xs:complexType>
</xs:schema>
```

Este *XML Schema* é o utilizado para gerar as classes através do *Binding Compiler* do *JAXB*.

A plataforma utilizada para compilação e utilização das classes *Java* geradas é o *IDE SunONE Studio 4 Update 1*. [SUN, 2003c]

No Anexo A descreve-se o modo como o *JAXB* gera classes por cada elemento que é indicado no *XML Schema*.

5 ARQUITECTURA DO ARQUIVO AUDIVISUAL

Neste capítulo é apresentada a arquitetura do sistema de arquivo audiovisual. Na figura 21 apresenta-se a arquitetura sobre a qual assenta o sistema. Esta arquitetura baseia-se no modelo clássico das três camadas: camada de dados, camada lógica e camada de apresentação.

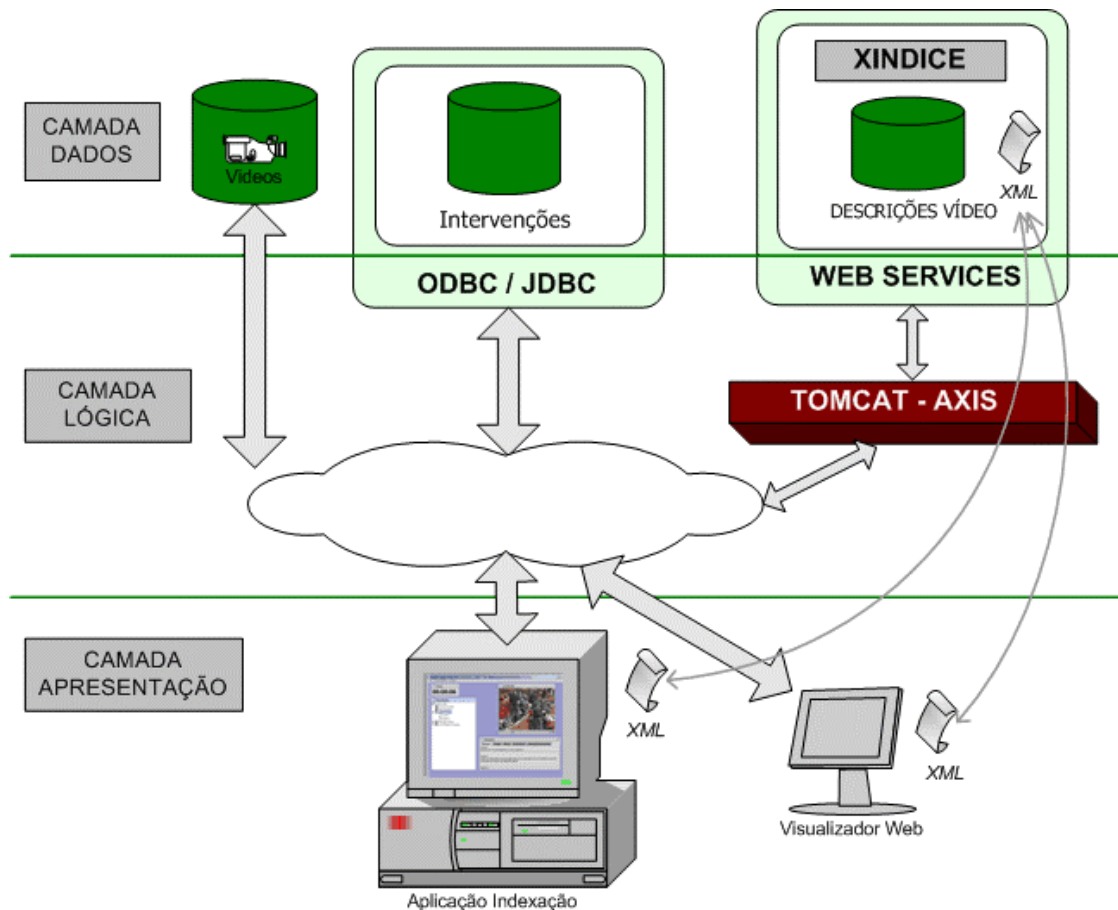


figura 21 - Arquitetura do sistema de arquivo audiovisual

5.1 CAMADA DE DADOS

A camada de dados é constituída por três componentes de armazenamento de informação:

1. Servidor de vídeos
2. Base de dados de Intervenções
3. Base de dados *XML* com descrições dos vídeos

Em seguida analisa-se cada um destes componentes em pormenor.

5.1.1 Servidor de vídeos

O primeiro componente da camada de dados é um repositório de informação que contém a colecção de vídeos pertencentes aos debates parlamentares da Assembleia da República.

Os vídeos encontram-se armazenados num servidor e estão organizados de acordo com uma estrutura hierárquica de modo a permitir a utilização de um método de descoberta automática dos mesmos.

O nome dos vídeos pode ser obtido através da expressão $S_{ns}L_{nl}SL_{nsl}N_{nsp}$, onde ns , nl , nsl , nsp , corresponde respectivamente ao número da série, legislatura, sessão legislativa e sessão parlamentar. Por exemplo, para o caso de um vídeo correspondente à sessão número 2 da 1.ª série, 8.ª legislatura 1.ª sessão legislativa o seu nome é S1L8SL1N2.

5.1.1.1 Conversão de vídeo analógico para digital

Originalmente os vídeos encontram-se armazenados em cassetes *VHS*. É necessário proceder à conversão dos mesmos para formato digital.

Este processo foi realizado utilizando um dispositivo que permitia capturar vídeo a partir de um videogravador analógico. Após a digitalização do vídeo foi efectuada a conversão para formatos de vídeo com compressão, como por exemplo o *Windows Media Video* e formato *MPEG I*.

Foram efectuados vários testes e apresenta-se de seguida uma tabela com as características da compressão de vídeo.

Formato	vídeo					áudio		tamanho	BitRate
	BitRate (Kbits/s)	codificação	Altura (píxeis)	Largura (píxeis)	fps	Sample rate (bits/s)	BitRate (bits/s)	(bytes)	Total (Kbits/s)
<i>Broadband-512</i>	448	wmv	320	240	25	44100	64	858433	512
<i>Broadband-340</i>	240	wmv	320	240	25	32000	32	583036	340
<i>HighQuality-320</i>	213	wmv	320	240	25	44100	88	503951	301
<i>MPEG I</i>	300	Mpeg I	320	240	25	44100	128	722361	428
<i>Dial-38</i>	30	wmv	160	120	15	8000	8	69951	38

Tabela 1 - Tabela de codificação de vídeo

Dos vários formatos testados optou-se por utilizar somente três. Um formato para permitir a visualização dos vídeos numa rede de banda larga, outro para permitir a visualização dos vídeo numa rede com uma largura de banda reduzida e outro formato para permitir a indexação dos vídeos.

O formato de vídeo utilizado para banda larga é o *HighQuality-320*. Como se pode observar na Tabela 1, este formato utiliza compressão *windows media video*, a imagem digitalizada tem uma altura de 320 píxeis e uma largura de 240 píxeis, ocupa uma largura de banda de 302 Kbps e tem uma cadência de 25 imagens por segundo.

O formato de vídeo utilizado para largura de banda reduzida é o *Dial-38*. Este formato utiliza também compressão *windows media video*, e a imagem digitalizada tem uma altura de 160 píxeis e uma largura de 120 píxeis, ocupa uma largura de banda de 38 Kbps e tem uma cadência de 15 imagens por segundo.

O formato de vídeo utilizado para a aplicação de indexação é o *MPEG I*, pois esta só suporta este formato de compressão de vídeo.

Este formato tem uma imagem digitalizada com altura de 320 píxeis e uma largura de 240 píxeis, ocupa uma largura de banda de 428 Kbps e tem uma cadência de 30 imagens por segundo.

5.1.2 Base de dados de Intervenções

O segundo componente da camada de dados é uma base de dados relacional que contém informação relativa às intervenções parlamentares dos oradores.

No conjunto dos sistemas de informação existentes na Assembleia da República, existe um sistema já implementado que possui as intervenções dos deputados. A partir desse sistema foi disponibilizado um conjunto de registos que permitiram criar uma base de dados.

Essa base de dados é relacional e encontra-se armazenada num sistema legado, disponibilizando informação relativa às intervenções parlamentares dos oradores nas várias sessões parlamentares.

Desta base de dados pode obter-se, entre outras, informações sobre o resumo, sumário, nome do orador e páginas onde se encontra a intervenção nos Diários da Assembleia da República.

Esta informação vai ser de extrema utilidade na construção da aplicação de indexação de vídeo, pois serve de guia para os utilizadores durante o processo de indexação do conteúdo dos vídeos.

5.1.3 Base de dados *XML* com descrições dos vídeos

O terceiro componente é uma base de dados *XML* que armazena as descrições dos vídeos. A base de dados *XML* com descrições vídeo tem como função armazenar as descrições dos vídeos indexados. O *SGBD* que gere esta informação é o *XIndex* [APACHE, 2003a].

O *XIndex* é uma base de dados que foi desenhada de raiz para armazenar documentos *XML*, daí que normalmente seja referida como sendo uma base de dados nativa *XML*.

A opção de utilizar uma base de dados nativa *XML* prende-se com o facto de, deste modo, não ser necessário mapear o documento *XML* para uma outra estrutura de armazenamento de dados relacional. A informação é inserida na base de dados como documentos *XML* e são devolvidos documentos *XML*.

Outra característica do *XIndex* é a sua flexibilidade, através da natureza semi-estruturada do *XML* e o modelo independente de *XML Schemas* que é utilizado. Este factores são importantes quando se armazenam estruturas *XML* muito complexas que seriam muito difíceis, ou mesmo impossíveis, de mapear em uma base de dados estruturada.

O *XIndex* utiliza *XPath* [W3C, 1999b] como linguagem de consulta e utiliza *XML:DB Xupdate* [LAUX, 2000] como linguagem de actualização. Dispõe também de uma implementação da *XML:DB API* para desenvolvimento de aplicações em *JAVA*. [APACHE, 2003a]

5.1.3.1 Organização dos registos *XML*

Para cada vídeo indexado existe um registo na base de dados, ou seja um ficheiro *XML*, que contém as informações acerca dos vários segmentos que fazem parte de um vídeo de uma sessão parlamentar. Os registos estão organizados de um modo hierárquico em colecções.

Através da ferramenta *XMLDB-GUI* [DSTC, 2003a] é possível observar o modo como estão organizados os registos na base de dados *XML*.

O *XMLDB-GUI* é uma aplicação *JAVA* baseada em componentes *Swing* que permite visualizar e modificar bases de dados que estão de acordo com a especificação *XML:DB API*. [DSTC, 2003b]

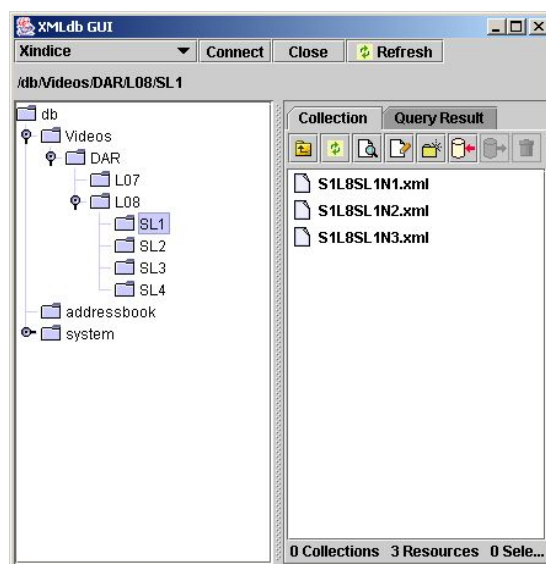


figura 22 - Organização dos registos na base de dados *XML*

As colecções dos vídeos encontram-se organizadas numa estrutura em árvore.

Analisando a figura 22 pode constatar-se que a colecção raiz tem por nome *db*. Seguidamente encontra-se a colecção *Videos* → *DAR*. Este nome serve para classificar e distinguir os documentos que fazem parte da colecção de vídeos dos Diários da Assembleia da República, pois podem existir mais colecções com outros conteúdos, como por exemplo os trabalhos das comissões parlamentares, das sessões especiais na sala de senado, etc.

Seguidamente encontram-se as colecções *L07* e *L08*. Estas indicam a legislatura a que dizem respeito os registos.

As colecções *SL1*, *SL2*, *SL3* e *SL4* servem para separar os registos por sessão legislativa.

5.2 CAMADA LÓGICA

Todos os componentes referidos na camada de dados assentam sobre a camada lógica. Nesta secção descrevem-se as tecnologias que foram utilizadas de modo a permitir a comunicação entre a camada de dados, onde se encontram os repositórios de informação e a camada de apresentação, onde se encontra a aplicação de indexação de vídeo e o visualizador *Web*.

Pretende-se que o sistema funcione de uma forma distribuída baseada no modelo de comunicações cliente-servidor e, para tal, é necessário, ao nível lógico, garantir a independência da camada de aplicação em relação à camada de dados onde se encontram os *SGBD's*.

5.2.1 Ligação à base de dados de intervenções

Na ligação à base de dados de intervenções através do visualizador *web* utiliza-se a tecnologia *ODBC (Open Database Connectivity)* [MICROSOFT, 2003a].

ODBC é uma interface de programação de aplicações (*API*) para base de dados com uma grande aceitação. É baseada na especificação *X/Open CAE "Data Management: SQL Call-Level Interface (CLI)"* e na especificação *ISO/IEC 9075-3:1995 (E) Call-Level Interface (SQL/CLI)*, e utiliza-se em *API's* de base de dados que usam *Structured Query Language (SQL)* [ANSI, 1996] como linguagem de acesso à base de dados. [MICROSOFT, 2003b]

Na ligação à base de dados de intervenções através da aplicação de indexação de vídeo utiliza-se a tecnologia *JDBC (JAVA Database Connectivity)* [SUN, 2003d].

A tecnologia *JDBC* é uma *API* que permite aceder a fontes de dados tabulares através da linguagem de programação *JAVA*. Disponibiliza conectividade entre diferentes sistemas de gestão de base de dados a uma vasta gama de bases de dados *SQL*.

Com a tecnologia *JDBC* um programador pode facilmente ligar a diferentes fontes de dados num ambiente heterogéneo. [SUN, 2003e]

5.2.2 Ligação à base de dados XML

O sistema de arquivo audiovisual utiliza *Web Services* [W3C, 2002] como meio de ligação à base de dados XML.

Os *Web Services* permitem estabelecer ligações com a base de dados XML de um modo distribuído, suportando o sistema vários utilizadores em simultâneo.

Outro aspecto importante da utilização dos *Web Services* na camada lógica é o nível de abstracção criado sobre o *SGBD* da base de dados XML. Garante-se assim que a manipulação dos registos da base de dados XML é feita de um modo independente do *SGBD XML*.

Mais tarde facilmente se troca o *SGDB XML* por outro diferente não havendo necessidade de reescrever o código de aplicações que acedam ao mesmo.

5.2.2.1 Modo de funcionamento

Um *Web Service* é um sistema de *software* identificado por um *URI* cuja interface e ligações públicas são definidas e descritas utilizando XML. A sua definição pode ser descoberta por outros sistemas de *software*. Estes sistemas podem depois interagir com o *Web Service* do modo descrito na sua definição, utilizando mensagens em XML encaminhadas por protocolos da Internet. [W3C, 2002]

5.2.2.2 Comunicação utilizando Web Services

A um nível conceptual pode-se olhar para o funcionamento dos *Web Service* como serviços disponibilizados via *Web*. [ARMSTRONG, 2003]

Sempre que é invocado um método disponibilizado num *Web Service* é enviada uma resposta para o cliente que fez o respectivo pedido. Este processo é semelhante ao pedido de uma página *Web*, em que quando um cliente efectua uma ligação a um servidor de *http* pedindo uma página *Web*, este responde enviando-lhe a respectiva informação em *HTML/XML*.

No caso dos *Web Services* a informação que circula na rede via *http*, é encapsulada utilizando um protocolo próprio que tem por nome *Simple Object Access Protocol (SOAP)*. [W3C, 2003] Este protocolo utiliza como linguagem de transporte de informação o *XML*. *SOAP* é um protocolo simples para intercâmbio de informação num ambiente descentralizado e distribuído. É um protocolo baseado em *XML* e possui três partes: um envelope que define uma arquitectura para descrever o que uma mensagem contém e como a processar, um conjunto de regras de codificação para expressar instâncias de tipos de dados definidos por aplicações, e uma convenção para representar chamadas e respostas a procedimentos remotos. [W3C, 2003]

Um dos grandes problemas das tecnologias *RPC* [SRINIVASAN, 1995], *COM* [MICROSOFT, 1995] e *DCOM* [BOX, 1998] é que, normalmente, é necessário efectuar configurações na *firewall* de modo a ser possível estabelecer a ligação entre o cliente e o servidor. Como os *Web Services* utilizam o *http* como meio de transporte e este serviço é utilizado pela maior parte dos servidores *Web*, normalmente, não é necessário fazer nenhuma configuração da *firewall*.

5.2.2.3 Clientes do Web Service

Os clientes do *Web Service* representam, neste caso concreto, a aplicação de indexação de vídeo e a página *Web* que apresenta os vídeos. Ambas efectuem ligações ao servidor do *Web Service*, que tem como função disponibilizar informação que se encontra armazenada na base de dados *XML*.

5.2.2.4 Servidor Web Service

Para que seja possível disponibilizar *Web Services* é mandatório dispor-se de um servidor *http*. Neste caso optou-se por utilizar o servidor *TOMCAT*, um servidor *open-source* da *Apache*. [APACHE, 2003b]

O *TOMCAT* é um repositório de *servlet's* para as tecnologias de *Java Servlets* e *JavaServer Pages*. As especificações das *Java Servlets* e *JavaServer Pages* são desenvolvidas pela *Sun* sob a orientação da *Java Community Process*. [APACHE, 2003b]

Este servidor por si só não permite a disponibilização de *Web Services*, daí a necessidade de instalar um *plugin* que permitisse disponibilizar os mesmos. O *plugin* utilizado tem por nome *AXIS*. [APACHE, 2003c]

O *Apache AXIS* é uma implementação *SOAP* do W3C. [APACHE, 2003c]

Os métodos que permitem manipular os documentos *XML* armazenados na base de dados *XML* são os seguintes:

1. *getDocument*
2. *getCollectionsCount*
3. *listCollections*
4. *createCollection*
5. *removeCollection*
6. *getDocumentCount*
7. *listDocuments*
8. *insertDocument*
9. *removeDocument*
10. *queryCollection*
11. *getLastQueryMoreResults*

A figura 23 apresenta os métodos disponibilizados pelo *Web Service* que efectua a ligação com a base de dados *XML*.

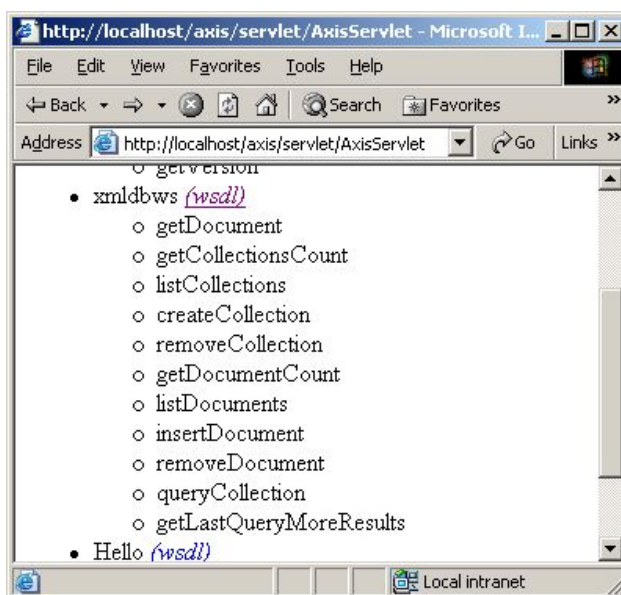


figura 23 - Métodos disponibilizados pelo *Web Service* de acesso à base de dados *XML*

O *Web Service* disponibilizado tem o nome de *xmlbws*.

Dos métodos disponibilizados pelo *Web Service* os mais relevantes para o desenvolvimento da aplicação de indexação de vídeo são os que permitem lidar directamente com os registos da base de dados *XML*.

O método *getDocument* permite obter um documento de uma colecção da base de dados *XML*.

O método *insertDocument* permite inserir um documento numa determinada colecção da base de dados *XML*.

O método *removeDocument* permite retirar um documento de uma determinada colecção da base de dados *XML*.

Finalmente, o método *listCollections* que permite obter uma listagem das sub-colecções existentes numa determinada colecção. Quando a base de dados *XML* ainda se encontra vazia ela contém somente a colecção *db*.

5.3 CAMADA DE APRESENTAÇÃO

Finalmente na camada de apresentação surge a aplicação de indexação, que serve como interface com o sistema de indexação, e o visualizador *Web*, que serve de interface com a generalidade dos utilizadores.

Dix et al. [DIX, 1998] classificam as propriedades de usabilidade de sistemas interactivos de acordo com os seguintes guias:

- Aprendizagem: previsão, sintético, familiaridade, generalidade, consistência
- Flexibilidade: diálogo intuitivo, multitarefa, robustez.

Durante a criação das interfaces tiveram-se em consideração estas regras de criação de interfaces humano-computador.

Devido ao nível de complexidade exigida para a criação quer da aplicação de indexação de vídeo quer do visualizador *Web* estes dois componentes são descritos em dois capítulos próprios.

6 APLICAÇÃO DE INDEXAÇÃO DE VÍDEO

A aplicação de indexação de vídeo tem como objectivo disponibilizar um ferramenta para técnicos que permita realizar o trabalho de indexação dos vídeos de um modo simples e eficaz. Através da utilização desta aplicação torna-se possível criar, alterar e eliminar descrições dos vídeos pertencentes a uma colecção de vídeos.

6.1 LINGUAGEM DE PROGRAMAÇÃO UTILIZADA

A escolha da linguagem de programação *JAVA*, que foi utilizada no desenvolvimento da aplicação de indexação, teve em consideração alguns aspectos que se pretendiam estudar durante o desenvolvimento do trabalho, nomeadamente a interoperabilidade entre aplicações. Outro aspecto que se teve em consideração durante a escolha foi o facto de muitos dos componentes referidos anteriormente, como é o caso do *JAXB*, do *XIndex*, o *TOMCAT* e o *AXIS*, já disporem de *API's* de programação em *JAVA*.

6.2 INTERFACE

A interface da aplicação de indexação de vídeo é um *MDI (Multiple Document Interface)* composto por quatro janelas internas. Cada uma das janelas tem uma funcionalidade específica associada. A figura 24 apresenta a interface da aplicação de indexação de vídeo.



figura 24 - Interface da aplicação de indexação de vídeo

Toda a informação apresentada é familiar para os utilizadores alvo e a informação é apresentada de um modo consistente.

Os diálogos apresentados na aplicação são intuitivos e, após algum treino, rapidamente os utilizadores aprendem a realizar as tarefas pretendidas.

6.2.1 Janela de tempo

A janela de tempo, apresentada na figura 25, mostra o tempo que decorreu do vídeo. Esta janela serve para o utilizador da aplicação de indexação de vídeo ter uma melhor percepção do instante de tempo do vídeo que se está a analisar. À medida que o utilizador vai visualizando o vídeo esta janela apresenta o tempo decorrido do mesmo. Permite também que o utilizador confirme se os tempos que se encontram registados para os segmentos de vídeo coincidem com o instante apresentado.



figura 25 - Janela de tempo

6.2.2 Janela de visualização do vídeo

A janela de visualização do vídeo, apresentada na figura 26, permite aos utilizadores visualizarem o vídeo que estão a indexar. Nesta janela, o operador dispõe de um conjunto de controlos que pode utilizar durante a visualização de um vídeo. Entre eles encontra-se um botão que permite ao utilizador iniciar ou parar a visualização do vídeo e uma barra de arrastamento que permite ao utilizador posicionar-se em diferentes instantes do vídeo. Os dois botões mais à direita servem respectivamente para fazer o ajuste do volume do som do vídeo e visualizar as características de codificação do vídeo.

Utilizou-se a *API* de *JAVA*, *Java Media Framework* [SUN, 2003f], para criar a janela onde é apresentado o vídeo. Como esta *API* apenas suporta o formato de vídeo *MPEG-I*, foi necessário criar um vídeo neste formato.

A *API Java Media Framework (JMF)* permite adicionar áudio, vídeo e outros tipos de media baseados no tempo, a aplicações ou *applets JAVA*. Este pacote opcional, que pode capturar e reproduzir múltiplos formatos de media, estende as capacidades de multimédia da plataforma *Java 2 Standard Edition, J2SE* [SUN, 2003g], e fornece aos programadores de aplicações multimédia um conjunto de ferramentas poderosas que permitem desenvolver aplicações escaláveis e multiplataforma. [SUN, 2003f]



figura 26 - Janela de visualização do vídeo

6.2.3 Janela de intervenções

Como se pode observar na figura 27, existe também uma janela que apresenta uma lista com as intervenções dos oradores para a sessão de vídeo que está a ser analisada.

Cada nodo da estrutura em árvore representa uma intervenções de um orador.

A um primeiro nível representam-se as intervenções dos oradores e a um segundo nível as sub-intervenções dos mesmos. As intervenções correspondem à totalidade do discurso do orador e as sub-intervenções correspondem a partes do discurso da intervenção do orador. Deste modo pode obter-se uma elevada granularidade na análise de um vídeo.

O ícone com a forma de um pedaço de filme com um traço em cima indica que a intervenção em causa ainda não foi indexada. Um ícone sem o traço indica que o sistema, pelo menos, já contém informação temporal para a intervenção em causa.

Através desta informação, o utilizador tem a percepção das intervenções que já foram indexadas e as que ainda terá que indexar no vídeo da sessão.

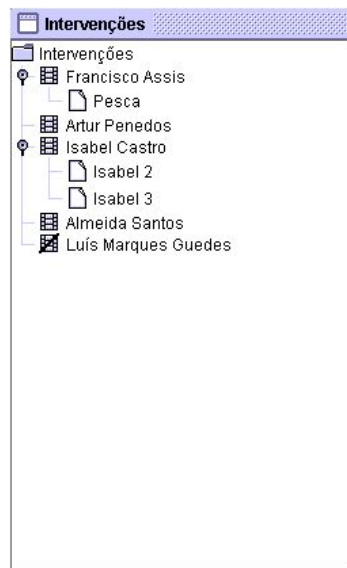


figura 27 - Janela de intervenções

6.2.4 Janela de anotações

A janela de anotações serve para visualizar e adicionar informação ao ficheiro *XML* com a descrição do vídeo. É constituída por um conjunto de classificadores que permitem ao utilizador seleccionar vários tipos de informação relativa ao vídeo.

Quando o utilizador selecciona uma intervenção na janela de intervenções aparece no classificador de Resumo o sumário, o resumo e as páginas do diário da Assembleia da República que fazem parte da intervenção seleccionada.

Através da leitura deste resumo e do sumário o utilizador fica com uma percepção dos temas que são tratados na intervenção que vai ser analisada e da sua extensão, pois se a intervenção aparecer em várias páginas é porque se trata de uma intervenção um pouco extensa. Esta informação é obtida automaticamente a partir da base de dados de intervenções, referida na secção 5.1.2. Na mesma janela o classificador de tempo, notas, descritores e anotação estruturada permitem ao utilizador visualizar e registar informação sobre o instante inicial, a duração de um segmento de vídeo, notas adicionais acerca da intervenção analisada, descritores e anotações estruturadas.

A figura 28 apresenta um exemplo da janela de intervenções com o sumário, o resumo e as páginas relativas a uma intervenção.

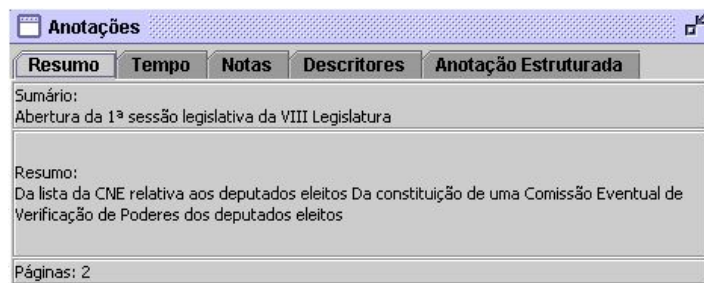


figura 28 - Janela de anotações com resumo de uma intervenção

6.2.5 Menu de acções e menu de configuração

O menu apresentado na figura 29 exemplifica as acções que o utilizador pode efectuar durante a indexação de um filme.

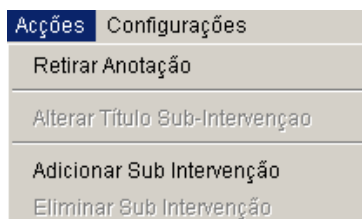


figura 29 - Menu de acções

As acções que são possíveis de realizar são disponibilizadas automaticamente pela aplicação. Se o utilizador estiver a indexar uma intervenção tem a possibilidade de retirar uma anotação previamente colocada ou adicionar uma sub-intervenção. Caso o utilizador esteja a indexar uma sub-intervenção tem a possibilidade de alterar o título da sub-intervenção ou eliminar a sub-intervenção que foi seleccionada. Todas as acções que o utilizador efectuar vão ter reflexo imediato no sistema de indexação.

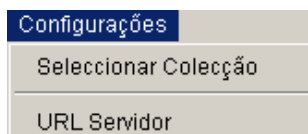


figura 30 - Configurações da aplicação

A figura 30 apresenta as configurações que podem ser feitas na aplicação de indexação. Inicialmente deve-se configurar o *URL* do servidor de *Web Services*. Este passo é essencial pois a comunicação com a base de dados *XML* é que vai permitir armazenar as descrições sobre os vídeos que estão a ser indexados. Após esta configuração torna-se possível listar as colecções que existem na base de dados, pois para obter esta listagem é invocado o método do *Web Service* que lista as colecções existentes na base de dados *XML*.

6.3 ARQUITECTURA DA APLICAÇÃO DE INDEXAÇÃO DE VÍDEO

A aplicação de indexação de vídeo permite interagir com o sistema de arquivo audiovisual. O seu principal objectivo é construir e manter documentos *XML* com as descrições dos vídeos que são armazenados na base de dados *XML*.

6.3.1 Ligação à base de dados XML

Como foi referido anteriormente o acesso à base de dados *XML* com as descrições dos vídeos é feito através de um *Web Service*. Para utilizar o *Web Service* é necessário ter conhecimento do *WSDL* (*Web Services Description Language*) [W3C, 2001b] que descreve o mesmo.

6.3.1.1 Descrição do Web Service

O *WSDL* é um formato *XML* para descrever os *Web Services* como um conjunto de terminais que manipulam mensagens que contêm informação orientada quer a documentos quer a procedimentos. [W3C, 2001b]

O *WSDL* é extensível para permitir descrições de terminais e das suas mensagens independentemente do formato da mensagem ou do protocolo de rede utilizado para comunicar. [W3C, 2001b]

A figura 31 apresenta o diagrama que ilustra os elementos que são apresentados num documento *WSDL* e como eles se encontram relacionados: [ORACLE, 2003]

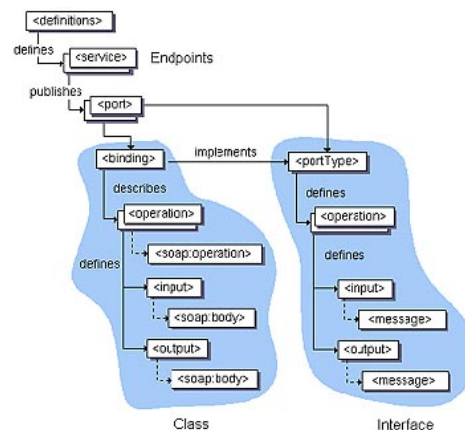


figura 31 - Diagrama de um documento WSDL [ORACLE,2003]

Um documento *WSDL* tem um elemento *definitions* que contém os elementos *types*, *message*, *operation*, *portType*, *binding*, *port* e *service* como se descreve na Tabela 2:

ELEMENTO	FUNCIONALIDADE
<i>definitions</i>	Define um ou mais serviços. Um elemento <i>definitions</i> suporta os seguintes atributos: <ul style="list-style-type: none"> • <i>name</i> – opcional • <i>targetNamespace</i> – é o <i>namespace</i> lógico para informação acerca deste serviço. Um documento <i>WSDL</i> pode importar outros documentos <i>WSDL</i> e atribuindo um único valor ao <i>targetNamespace</i> garante que os <i>namespaces</i> não colidem. • <i>xmlns</i> – é o <i>namespace</i> por defeito do documento <i>WSDL</i> e tem o valor de http://schemas.xmlsoap.org/wsdl/. Todos os elementos de um documento <i>WSDL</i>, como por exemplo <definitions>, <types> e <message>, residem neste <i>namespace</i>. • <i>xmlns:xsd</i> e <i>xmlns:soap</i> – são definições para <i>namespaces</i> standard que são utilizados para especificar informação específica do SOAP e também tipos de dados. • <i>xmlns:ts</i> – representa o <i>namespace this</i>
<i>types</i>	Disponibiliza informação sobre qualquer tipo de dados complexo utilizado no documento <i>WSDL</i> . Quando se utilizam só tipos de dados simples não é necessária a utilização deste elemento.
<i>message</i>	Uma descrição abstracta da informação que vais ser enviada
<i>operation</i>	Uma descrição abstracta da acção suportada pelo serviço
<i>portType</i>	Um conjunto abstracto do conjunto de operações suportadas por um ou mais terminais
<i>binding</i>	Descreve como a operação é invocada através da especificação do protocolo concreto e especificação do formato de dados para as operações e para as mensagens.
<i>port</i>	Especifica um único terminal como um endereço para a ligação, logo define um único terminal para a comunicação.
<i>service</i>	Especifica o(s) endereço(s) do porto para a ligação. O serviço é uma colecção de terminais e portos da rede.

Tabela 2 - Descrição dos elementos de um documento WSDL [ORACLE, 2003]

Um programador ou ferramentas de desenvolvimento automáticas podem criar ficheiros *WSDL* para descrever um serviço e podem colocar a descrição disponível através da Internet. Programadores e ferramentas de desenvolvimento do lado do cliente podem depois utilizar as descrições *WSDL* publicadas para obter informação disponível sobre o serviço e para construir e criar *proxies* ou *templates* de programas que acedem a serviços disponíveis. [ORACLE, 2003]

O *WSDL* que descreve o *Web Service* utilizado pela aplicação de indexação de vídeo pode ser obtido através do endereço <http://tribunal.ieeta.pt/axis/services/xmldbws?wsdl> que é onde se encontra o sistema com o *Web Service*, como se pode observar na figura 32.

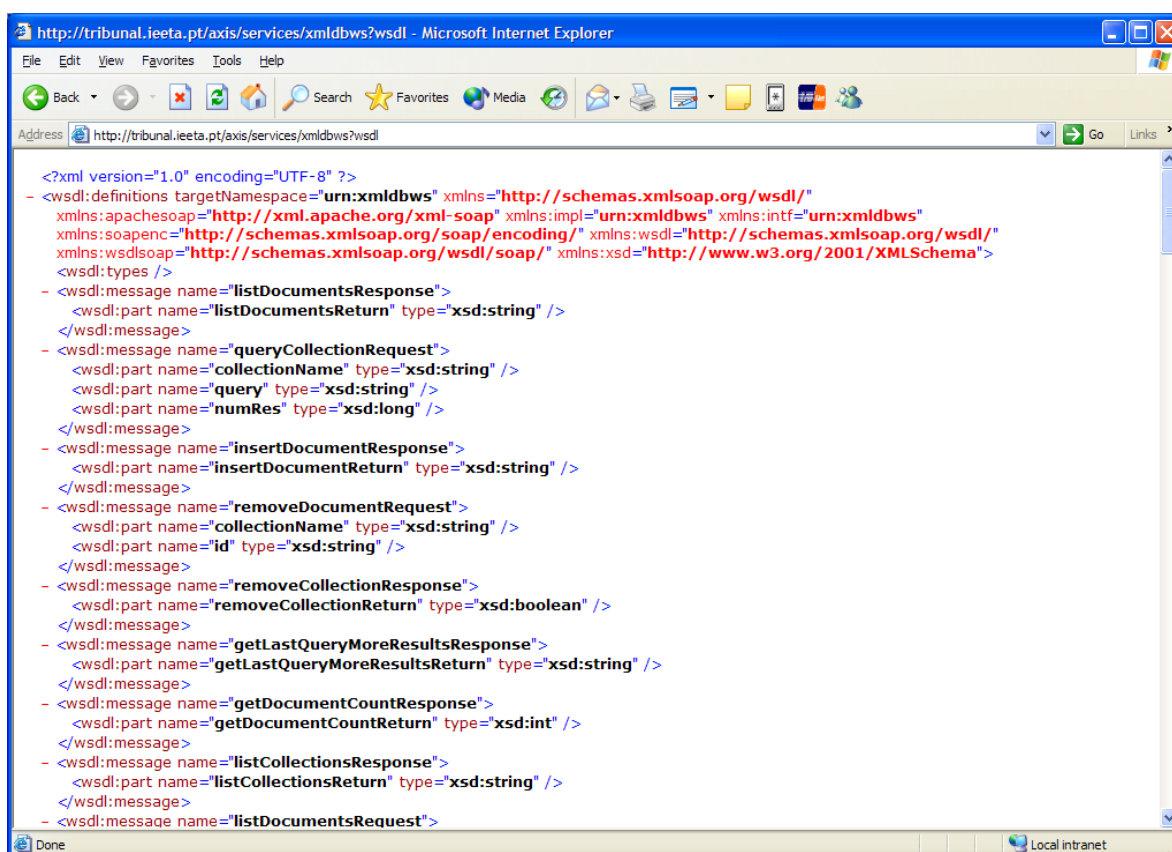


figura 32 - Descrição *WSDL* do *Web Service*

6.3.1.2 Criação de um cliente na aplicação para aceder ao *Web Service*

O *WSDL* com a descrição do serviço é depois utilizado numa ferramenta que faz parte do *AXIS* com o nome *WSDL2JAVA* [APACHE, 2003d]. Esta ferramenta serve para gerar um conjunto de classes *Java* que permitem efectuar a ligação ao servidor do *Web Service*.

Após efectuar a compilação do ficheiro *WSDL* que descreve o serviço são criadas quatro classes *JAVA* que permitem construir um cliente para o *Web Service*.

As classes que foram geradas são as seguintes:

1. *IXmldbws.java* - interface para o serviço gerado.
2. *XmldbwsService.java* - interface do serviço para o lado do cliente.
3. *XmldbwsServiceLocator.java*- implementação do serviço do lado do cliente.
4. *XmldbwsSoapBindingStub.java*- *stub* para o lado do cliente.

Estas classes ficam organizadas dentro de um *pacote* que depois é compilado e adicionado à aplicação de indexação de vídeo, ficando assim disponíveis os métodos oferecidos pelo *Web Service* na aplicação de indexação.

6.4 PROCESSO DE CRIAÇÃO DE UMA DESCRIÇÃO VÍDEO

O processo de criação de uma descrição de um vídeo passa por várias fases. Em seguida descrevem-se os passos necessários para a criação de uma descrição de um vídeo e o modo como a informação dos vários sistemas que se encontram na camada de dados é apresentada ao utilizador. Parte-se do princípio que já foi configurado o *URL* do servidor do *Web Service* e a colecção que está a ser indexada.

Inicialmente o utilizador começa por indicar qual é o novo filme que deseja indexar. Após seleccionar o filme que vai ser indexado, aparece a janela com o vídeo seleccionado e surge também uma janela com as propriedades do filme, como se pode observar na figura 33.



figura 33 - Propriedades de um vídeo

A janela de propriedades tem como objectivo disponibilizar uma interface que permita descrever as características do vídeo que está a ser analisado, nomeadamente das características de índole geral, codificação de vídeo e áudio e notas textuais sobre o vídeo. Estas propriedades são obtidas automaticamente analisando o vídeo. Após carregar no botão *OK* é iniciado um conjunto de acções que vão permitir realizar as anotações. É efectuada uma ligação à base de dados de intervenções para obter informações relativa às intervenções dos vários oradores e é criado um novo documento na base de dados *XML* com as descrições.

Inicialmente, este documento só contém as descrições sobre as propriedades de vídeo e não contém nenhuma decomposição temporal. Os segmentos de vídeo vão sendo inseridos à medida que vai sendo feita a indexação do conteúdo audiovisual. A figura 34 apresenta o documento *XML* que é criado inicialmente. Como se pode observar este documento contém somente informação relativa às propriedades do vídeo e ainda não têm qualquer informação relativa à segmentação do vídeo.

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<Mpeg7>
  <Description>
    <MultimediaContent>
      <AudioVisual>
        <MediaInformation>
          <Header>
            <CreationTime>26-06-2003</CreationTime>
          </Header>
          <MediaProfile master="true">
            <MediaFormat>
              <Content>
                <Name>audiovisual</Name>
              </Content>
              <Medium>
                <Name>hardisk</Name>
              </Medium>
              <FileFormat>
                <Name>MPEG</Name>
              </FileFormat>
              <FileSize>24897540</FileSize>
              <System>
                <Name>PAL</Name>
              </System>
              <Bandwidth>512.0</Bandwidth>
              <BitRate>428</BitRate>
              <VisualCoding>
                <Format colorDomain="color">
                  <Name>MPEG</Name>
                </Format>
                <Frame aspectRatio="1.0" width="320" height="240"></Frame>
              </VisualCoding>
              <AudioCoding>
                <Format>
                  <Name>MPEG Audio</Name>
                </Format>
                <AudioChannels>2</AudioChannels>
                <Sample rate="44100"></Sample>
              </AudioCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier>S1L8SL1N11</InstanceIdentifier>
              <MediaLocator>
                <MediaUri>http://parlamento.ieeta.pt/S1L8SL1N11</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
        </MediaInformation>
        <TextAnnotation>
          <FreeTextAnnotation>Notas sobre o filme</FreeTextAnnotation>
        </TextAnnotation>
        <MediaTime>
          <MediaTimePoint>00:00:00</MediaTimePoint>
          <MediaDuration>00:02:22</MediaDuration>
        </MediaTime>
        <TemporalDecomposition></TemporalDecomposition>
      </AudioVisual>
    </MultimediaContent>
  </Description>
</Mpeg7>

```

figura 34 - Descrição XML inicial do vídeo

Após ser criado o documento na base de dados *XML* a interface da aplicação de indexação de vídeo adapta-se para permitir a inserção de anotações relativas ao conteúdo audiovisual apresentando o aspecto da figura 35.



figura 35 - Criação de uma descrição

A partir deste ponto o utilizador inicia o processo de indexação propriamente dito e para tal tem de observar o vídeo, registar os tempos iniciais e finais da intervenção e registar possíveis palavras chave, descritores, que se enquadrem com o discurso do orador.

Como exemplo vai ser utilizada a indexação de uma intervenção do deputado Francisco Louçã, onde este orador discursa sobre a Organização Mundial do Comércio e coloca duas perguntas ao Ministro das Finanças e da Economia Pina Moura.

O utilizador começa por visualizar o vídeo até verificar que o deputado vai realizar uma intervenção. Selecciona então o nome do deputado na janela das intervenções. Aparece automaticamente na janela de anotações o sumário, o resumo e uma indicação com as páginas do diário da Assembleia da República onde aparece a intervenção, como se pode observar na figura 36.

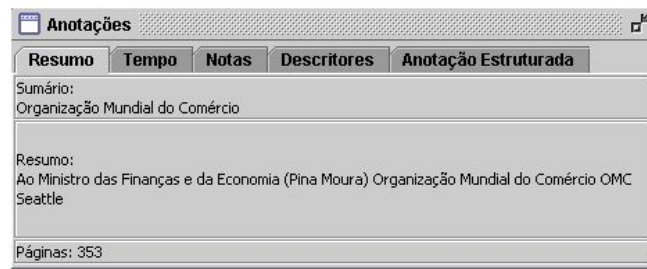


figura 36 - Janela de anotações da intervenção do deputado Francisco Louçã

Quando o deputado inicia a sua intervenção o utilizador pára o vídeo e, na janela de anotações, muda para o classificador Tempo. Regista o tempo inicial e continua a visualizar o vídeo até o deputado terminar o seu discurso. Nesse instante pára novamente o vídeo e regista a duração da intervenção do deputado. Note-se que estas anotações de tempo são feitas simplesmente através do premir de um botão na interface, não há necessidade de escrever o tempo em nenhuma caixa de texto. A figura 37 ilustra a situação mencionada.



figura 37 - Janela de anotações com o classificador do tempo

Após registada a duração da intervenção do deputado, automaticamente, o ícone com a indicação de que o segmento de vídeo já foi indexado muda de imagem e é inserida uma anotação no documento *XML* que contém as descrições do vídeo.

Analisando o documento *XML* com a descrição do conteúdo audiovisual pode-se observar que, dentro do elemento *TemporalDecomposition*, é inserido um elemento *AudioVisualSegment*, com o *ID* igual ao *ID* do registo da base de dados de intervenções.

É adicionado também um elemento *MediaTime* que contém os elementos *MediaRelTimePoint*, com a indicação do instante inicial do segmento de vídeo, e o elemento *MediaDuration*, com a indicação da duração do segmento de vídeo.

```

<TemporalDecomposition>
  <AudioVisualSegment id="313">
    <MediaTime>
      <MediaRelTimePoint>00:00:06</MediaRelTimePoint>
      <MediaDuration>00:02:16</MediaDuration>
    </MediaTime>
  </AudioVisualSegment>
</TemporalDecomposition>

```

Durante a intervenção, o utilizador repara que o deputado menciona que ia fazer duas perguntas ao ministro das Finanças e Economia. Acha o facto relevante e decide colocar uma nota na intervenção para que essa fique registada. O classificador de Notas na janela de anotações permite-lhe adicionar notas ao segmento de vídeo. Basta ao utilizador escrever a nota e, em seguida, registá-la, como se pode observar na figura 38.

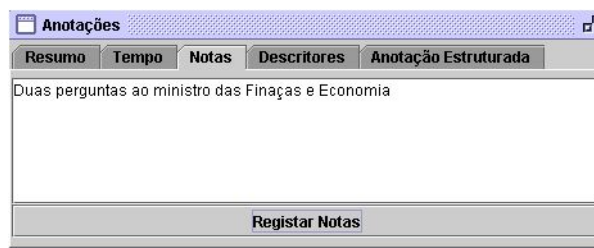


figura 38 - Registo de notas numa intervenção

Esta nota fica armazenada dentro de um elemento *FreeTextAnnotation* que por sua vez se encontra dentro do elemento *TextAnnotation*.

```

<TemporalDecomposition>
  <AudioVisualSegment id="313">
    <TextAnnotation>
      <FreeTextAnnotation>Duas perguntas ao ministro das Finanças e
Economia</FreeTextAnnotation>
    </TextAnnotation>
    <MediaTime>
      <MediaRelTimePoint>00:00:06</MediaRelTimePoint>
      <MediaDuration>00:02:16</MediaDuration>
    </MediaTime>
  </AudioVisualSegment>
</TemporalDecomposition>

```

A janela de anotações possui um classificador destinado aos descritores relativos à intervenção do deputado. Neste caso concreto adicionou-se o descritor OMC – Organização Mundial do Comércio e o descritor Agenda. Note-se que a adição de descritores obedece a um dicionário de termos controlado. No caso da Assembleia da República é utilizado o *Thesauros Eurovoc* [EUROPEAN, 2003]. A figura 39 apresenta os descritores adicionados.

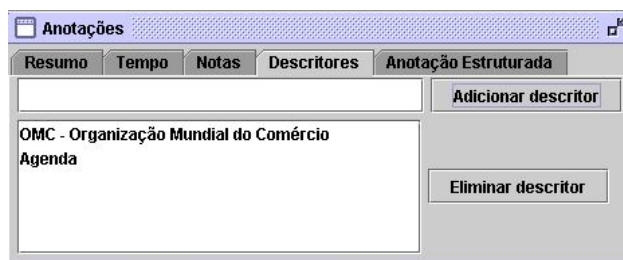


figura 39 - Registo de descritores numa intervenção

Estes descritores ficam armazenados dentro do elemento *KeywordAnnotation* estando cada descritor propriamente dito armazenado num elemento *Keyword*.

```
<TemporalDecomposition>
  <AudioVisualSegment id="313">
    <TextAnnotation>
      <FreeTextAnnotation>Duas perguntas ao ministro das Finanças e
Economia</FreeTextAnnotation>
      <KeywordAnnotation>
        <Keyword> OMC - Organização Mundial do Comércio</Keyword>
        <Keyword>Agenda</Keyword>
      </KeywordAnnotation>
    </TextAnnotation>
    <MediaTime>
      <MediaRelTimePoint>00:00:06</MediaRelTimePoint>
      <MediaDuration>00:02:16</MediaDuration>
    </MediaTime>
  </AudioVisualSegment>
</TemporalDecomposition>
```

Após concluído este processo de indexação a intervenção do deputado fica devidamente indexada e é possível utilizar sistemas de pesquisa para pesquisar a informação que foi associada à intervenção.

7 VISUALIZADOR WEB

7.1 INTRODUÇÃO

Depois de apresentada a ferramenta de indexação de vídeo apresenta-se neste capítulo o visualizador *Web*. Este visualizador é a interface que permite a vários utilizadores visualizarem as intervenções dos oradores em formato vídeo. O visualizador *Web* está inserido no sistema de Diários da Assembleia da República Electrónicos.

7.2 MODO DE FUNCIONAMENTO

Desde o início do desenvolvimento do sistema que o grande tema associado aos *Web Services* era a interoperabilidade.

Como a aplicação de indexação de vídeo foi desenvolvida na linguagem de programação *JAVA*, que faz parte da plataforma *SUN*, e o visualizador *Web* foi desenvolvido numa linguagem de programação que faz parte da plataforma *Microsoft*, foi possível observar duas aplicações a funcionarem em plataformas distintas e a trocarem informação entre si.



figura 40 - Esquema de funcionamento do *Web Service*

Na figura 40 observa-se o esquema de funcionamento do *Web Service* no que diz respeito à comunicação entre o cliente *Web* e a base de dados *XML* com as descrições dos vídeos.

Como foi referido na secção 5.1.3 a base de dados *XML* foi implementada em *JAVA*. O servidor do *Web Service*, *AXIS*, também foi desenvolvido em *JAVA* e implementa um conjunto de métodos que permitem aceder e manipular os registos da base de dados *XML*.

Aparece também na figura 40 uma *firewall* com uma seta que a atravessa de um lado ao outro. Esta parte da imagem pretende representar a facilidade de utilização dos *Web Services*, uma vez que, normalmente, não é necessário configurar a *firewall* pois o porto de comunicação utilizado para estabelecer a comunicação é o porto 80, correspondente ao serviço *http*.

No lado esquerdo da figura 40 aparece o cliente do *Web Service*, que neste caso concreto do arquivo audiovisual representa o serviço *IIS (Internet Information Services)* da Microsoft. Este serviço acede à base de dados *XML* através do *Web Service* e apresenta aos clientes *Web* a informação dos segmentos de vídeo que estão indexados para um vídeo de uma sessão parlamentar.

A interface de visualização dos vídeos é uma *aspx* e foi desenvolvida em *C#* [MICROSOFT, 2003c] com o *IDE Visual Studio .NET* [MICROSOFT, 2003d].

Adicionando uma referência *web*, criou-se um cliente do *Web Service* que permite efectuar a ligação à base de dados *XML*. Como se pode verificar na figura 41, os *Web Services* permitem obter interoperabilidade entre uma aplicação desenvolvida em *JAVA* na plataforma *Sun* e outra aplicação desenvolvida em *C#* na plataforma *Microsoft*.

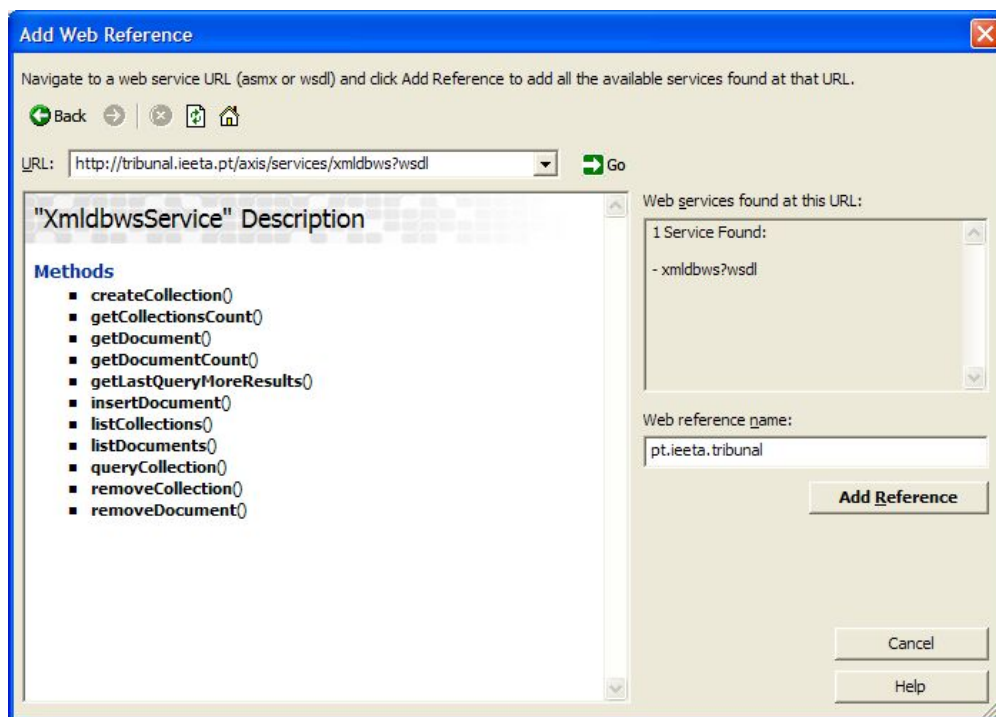


figura 41 - Referência Web para o Web Service xmldbws

7.3 VISUALIZAÇÃO DE UM VÍDEO

Através do visualizador *Web* um utilizador pode visualizar quer um vídeo completo quer segmentos de vídeo de uma sessão parlamentar. A opção de visualizar um vídeo surge depois de o utilizador efectuar uma pesquisa no sistema dos Debates Parlamentares. Após efectuar a pesquisa é apresentado o texto do diário e, na mesma página, o utilizador tem a hipótese de seleccionar um *link* que lhe permite visualizar o vídeo da Sessão.

A figura 42 apresenta uma página do sistema Debates com o link para visualizar o vídeo.



figura 42 - *Link* para seleccionar a opção de visualizar o vídeo da sessão

A figura 43 permite observar a interface que foi desenvolvida para visualizar os vídeos. Esta interface é constituída por uma estrutura em árvore, que indica quais as intervenções dos oradores no decorrer da sessão, e uma janela para visualizar os vídeos. As intervenções que têm um ícone com um traço por cima são aquelas que ainda não se encontram indexadas. As intervenções sem o traço já estão indexadas e é portanto possível observar o segmento de vídeo que diz respeito às mesmas. Se houver sub-intervenções, estas aparecem como elementos “filhos” da intervenção.

Na figura 43 pode-se também observar que a intervenção do orador Francisco Louça já se encontra indexada e que esta intervenção tem duas sub-intervenções: uma com o título “*primeira pergunta*” e outra com o título “*Segunda pergunta*”.

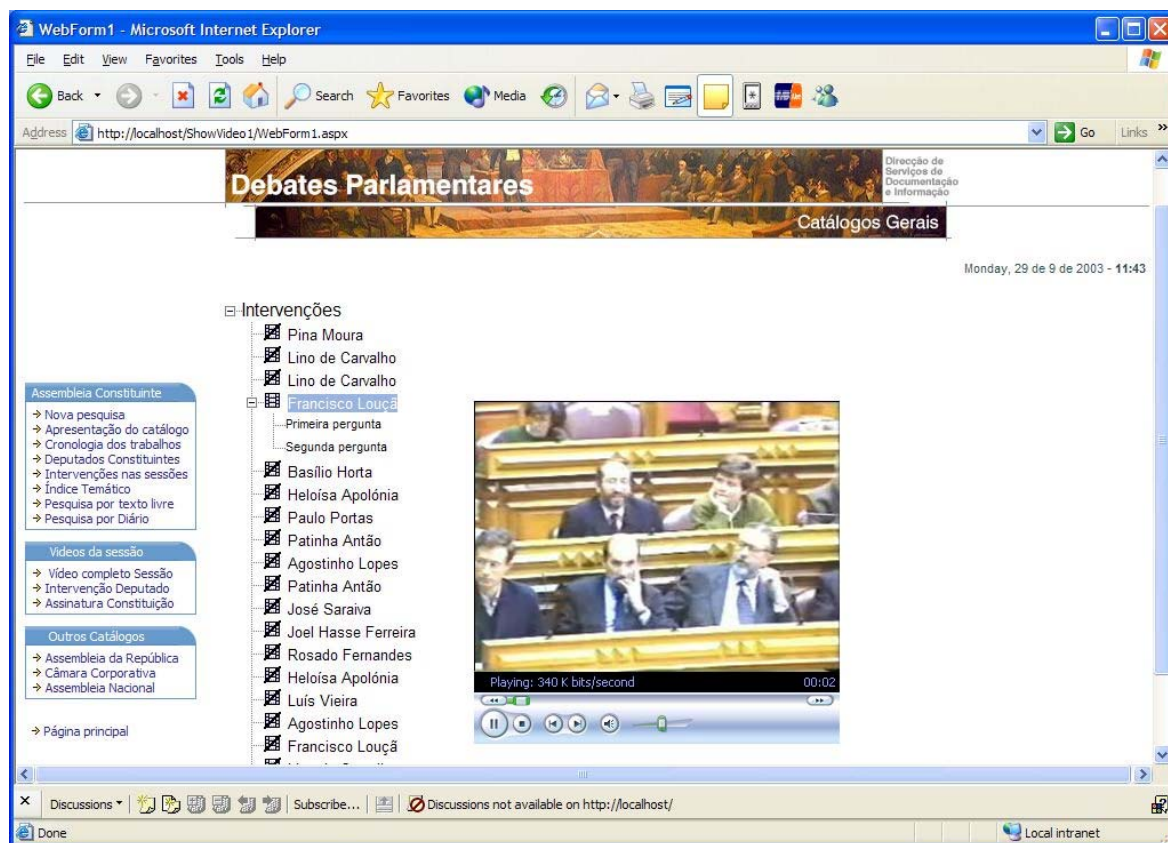


figura 43 - Interface Web de visualização dos vídeos

8 DESEMPENHO DA BASE DE DADOS XML *XINDEXE*

8.1 INTRODUÇÃO

Este capítulo tem como objectivo analisar o desempenho da base de dados *XML* nativa *XIndexe* e disponibilizar um conjunto de informação que pode ser preponderante na escolha deste sistema de gestão de base de dados para armazenamento e disponibilização de informação no formato *XML*.

Os resultados deste estudo podem ser utilizados na concepção e desenvolvimento de outros sistemas que utilizem *XML*.

O *XML* é, neste momento, a linguagem de descrição por excelência da Internet. Basta olhar para os grandes fabricantes de *software* como a *Microsoft* ou a *Sun* para verificar a quantidade de ferramentas que permitem manipular informação no formato *XML*.

Esta linguagem está associada a várias áreas dos sistemas de informação e aparece em sistemas de armazenamento de dados assim como em tecnologias que visam permitir a transferência de informação entre aplicações.

8.2 METODOLOGIA

Para realizar o estudo criaram-se 50.000 documentos *XML* com descrições de conteúdos audiovisuais de acordo com a estrutura descrita na secção 4.2.4.

De uma forma resumida, esta estrutura é composta por um conjunto de elementos descritivos do formato de codificação áudio e vídeo e por um conjunto de elementos *AudioVisualSegment* que têm como funcionalidade associada armazenar informação temporal e anotações sobre os segmentos de vídeo que compõem a totalidade de um filme.

Tendo em conta que um filme pode conter vários segmentos de vídeo, o número de elementos *AudioVisualSegment* que existem nos documentos foi criado de um forma aleatória, podendo cada documento conter entre um e dez elementos *AudioVisualSegment*.

Para cada segmento de vídeo foram associados um conjunto de descritores. O número de descritores para cada segmento de vídeo foi calculado também de forma aleatória a variar

entre 1 e 25. Cada descritor é seleccionado de uma forma aleatória a partir de um dicionário controlado com diferentes descritores. Os descritores são armazenados em elementos *Keyword* dentro do elemento *AudioVisualSegment*. Após a criação dos documentos que servem para teste efectuaram-se simulações com blocos de 10.000 documentos para poder analisar o efeito de carga na base de dados. Foram efectuados dois tipos de testes:

- Analisaram-se as acções de:
 - inserir um documento
 - obter um documento
 - remover um documento da base de dados *XML*.
- Efectuaram-se testes nas pesquisas em documentos existentes na base de dados *XML*

Pesquisou-se dentro da base de dados por um documento com um *AudioVisualSegment* com um determinado *ID* e por documentos que contivessem descritores com determinadas palavras. As pesquisas foram efectuadas antes e depois da criação de índices para observar o efeito da indexação.

Os tempos foram medidos num *Pentium III* 600 Mhz, com 512 Mb de *Ram*. Mantiveram-se as mesmas condições para efectuar as simulações de modo a obter o diferencial de tempo entre a utilização de *Web Services* e o acesso directo à base de dados *XML*. O servidor dos *Web Services* estava instalado na mesma máquina onde se encontrava a base de dados *XML* para minimizar os atrasos sofridos na rede.

Criou-se uma aplicação que permite realizar as tarefas mencionadas acima e que facilita a tarefa de análise. O único aspecto que não foi automatizado foi a criação dos índices.

Na aplicação pode seleccionar-se uma opção para efectuar os testes utilizando *Web Services*. Caso essa opção não seja seleccionada o acesso à base dados é feito através do *driver XML:DB API*. A figura 44 apresenta a interface da aplicação criada.



figura 44 - Interface da aplicação de análise

Esta aplicação mede os tempos e regista-os em ficheiros de *log* que servem posteriormente para analisar os resultados.

8.3 APRESENTAÇÃO DOS RESULTADOS

Tal como referido, a apresentação dos resultados será feita de duas formas distintas. Primeiro analisa-se o acesso à base de dados *XML* utilizando o *driver XML:DB API*. Depois apresentam-se os resultados do acesso à base de dados *XML* utilizando *Web Services*. Finalmente, apresentam-se os resultados dos efeitos da indexação na base de dados. A partir dos ficheiros de *log* que armazenam a informação com os tempos de acesso à base de dados *XML* construíram-se gráficos para visualizar melhor os resultados.

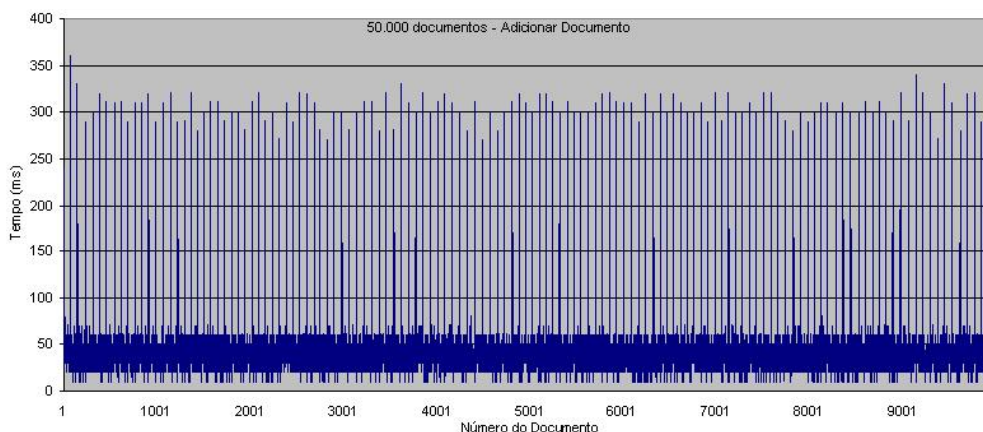
8.3.1 Acesso através do *driver XML:DB API*

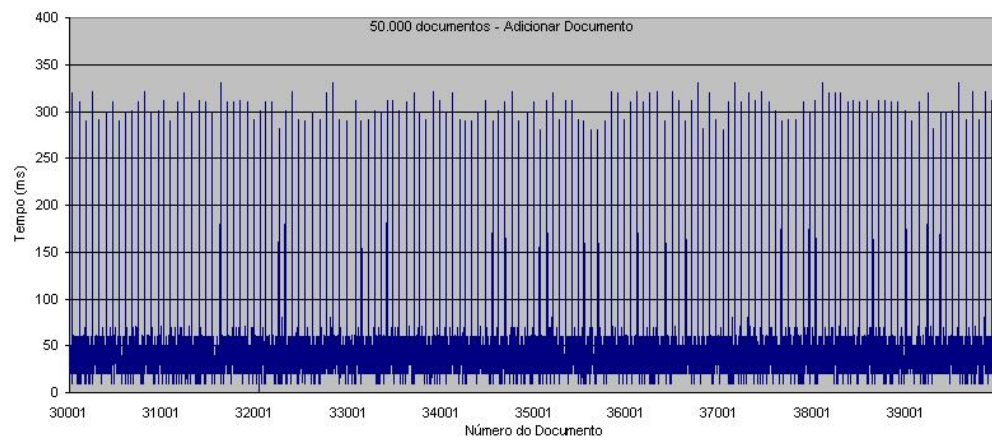
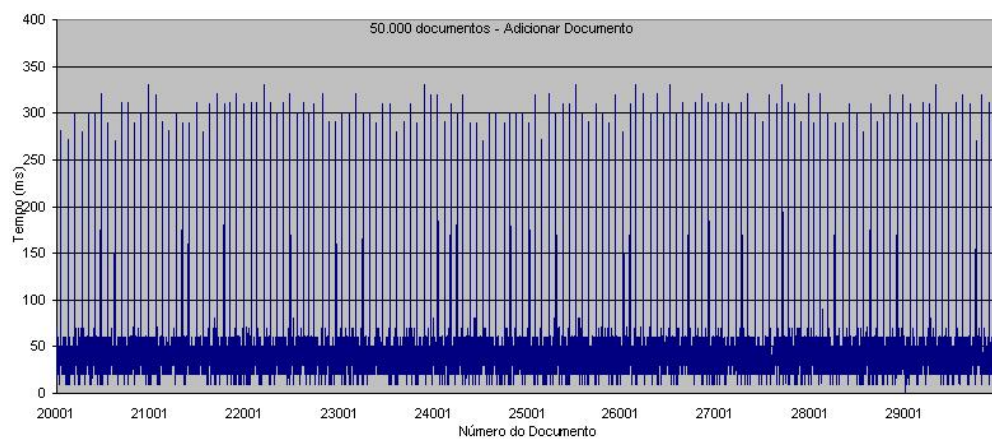
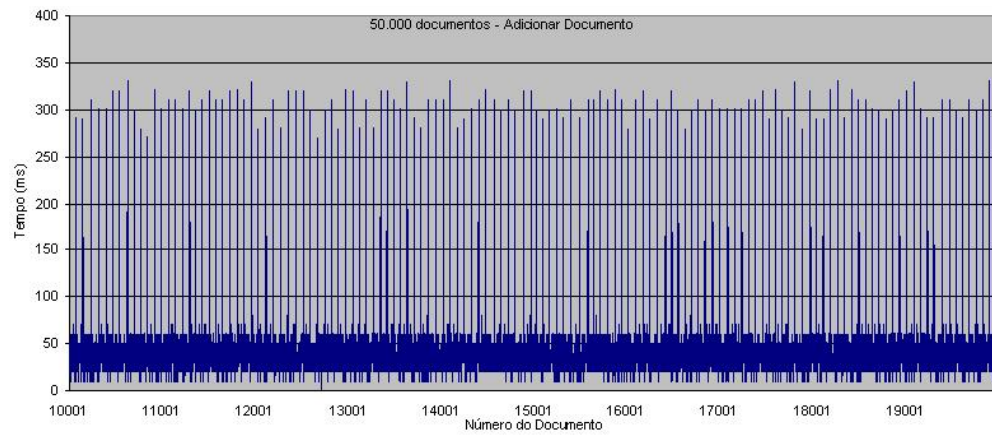
Os primeiros testes que se efectuaram para analisar o comportamento da base de dados foi através de uma ligação utilizando o *driver XML:DB API*.

8.3.1.1 Adicionar documentos

Apresentam-se de seguida os gráficos com o tempo despendido a adicionar um documento à base de dados *XML* e uma tabela com os tempos máximos e tempos médios para realizar esta operação. Para os testes com um grande número de documentos repartiu-se a apresentação por blocos de 10.000 documentos para visualizar melhor os resultados.

A. 50.000 documentos





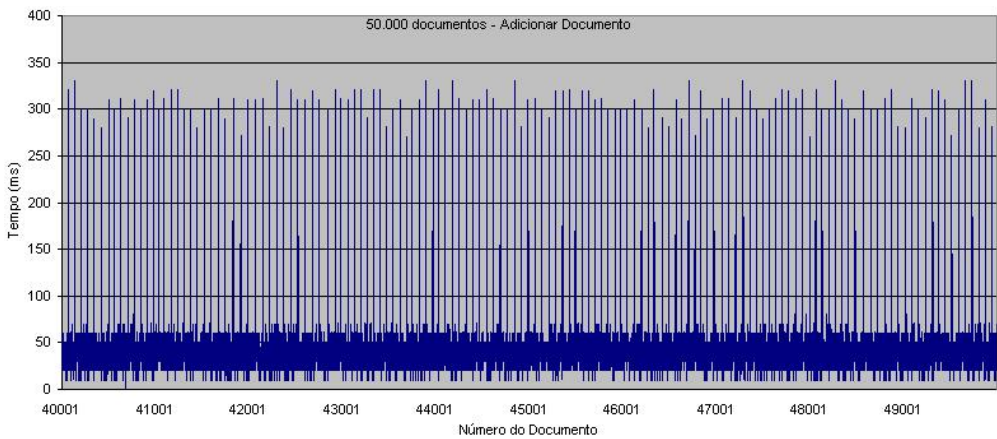
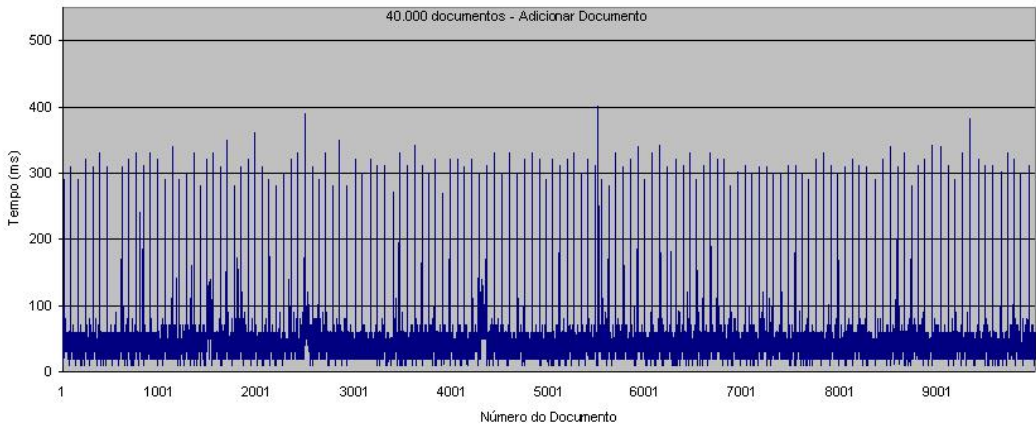


figura 45 - Gráficos de tempo despendido ao adicionar 50.000 documentos

<i>tempos registados (ms)</i>	
média	42
máximo	361

Tabela 3 - Tempos registados ao adicionar 50.000 documentos

B. 40.000 documentos



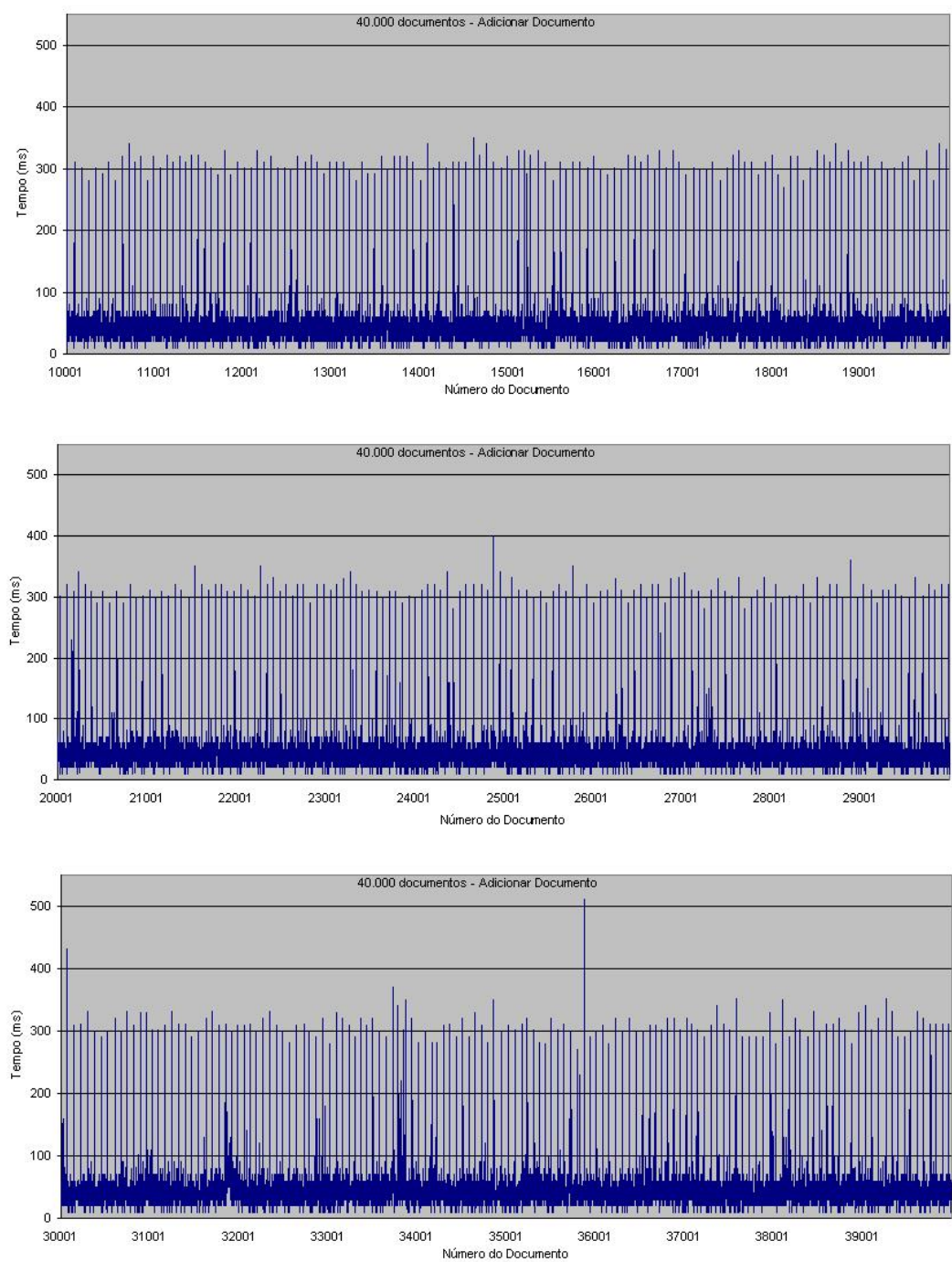


figura 46 - Gráficos do tempo despendido ao adicionar 40.000 documentos

<i>tempos registados (ms)</i>	
média	44
máximo	511

Tabela 4 - Tempos registados ao adicionar 40.000 documentos

C. 30.000 documentos

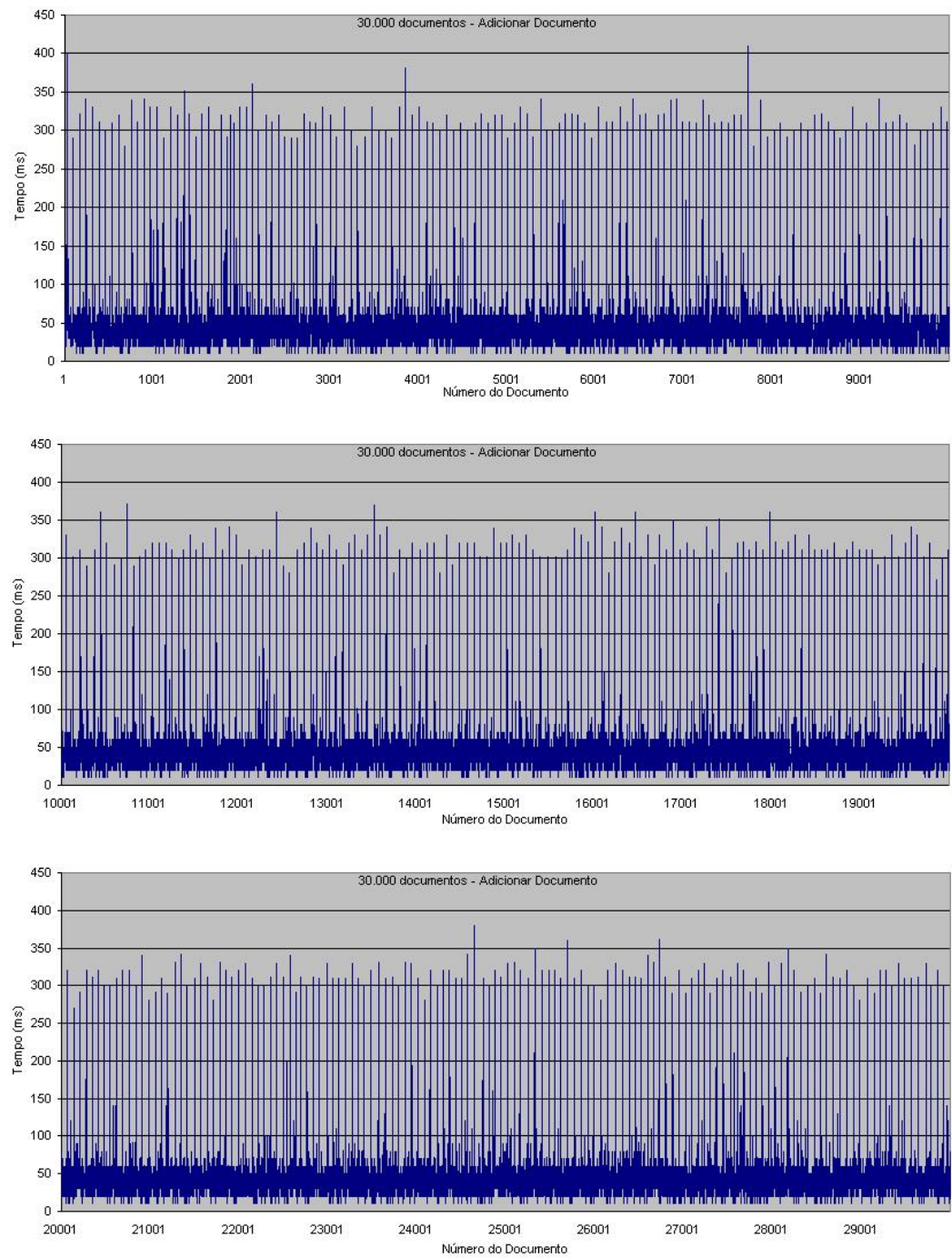


figura 47 - Gráficos do tempo despendido ao adicionar 30.000 documentos

<i>tempos registados (ms)</i>	
média	45
máximo	410

Tabela 5 - Tempos registados ao adicionar 30.000 documentos

D. 20.000 documentos

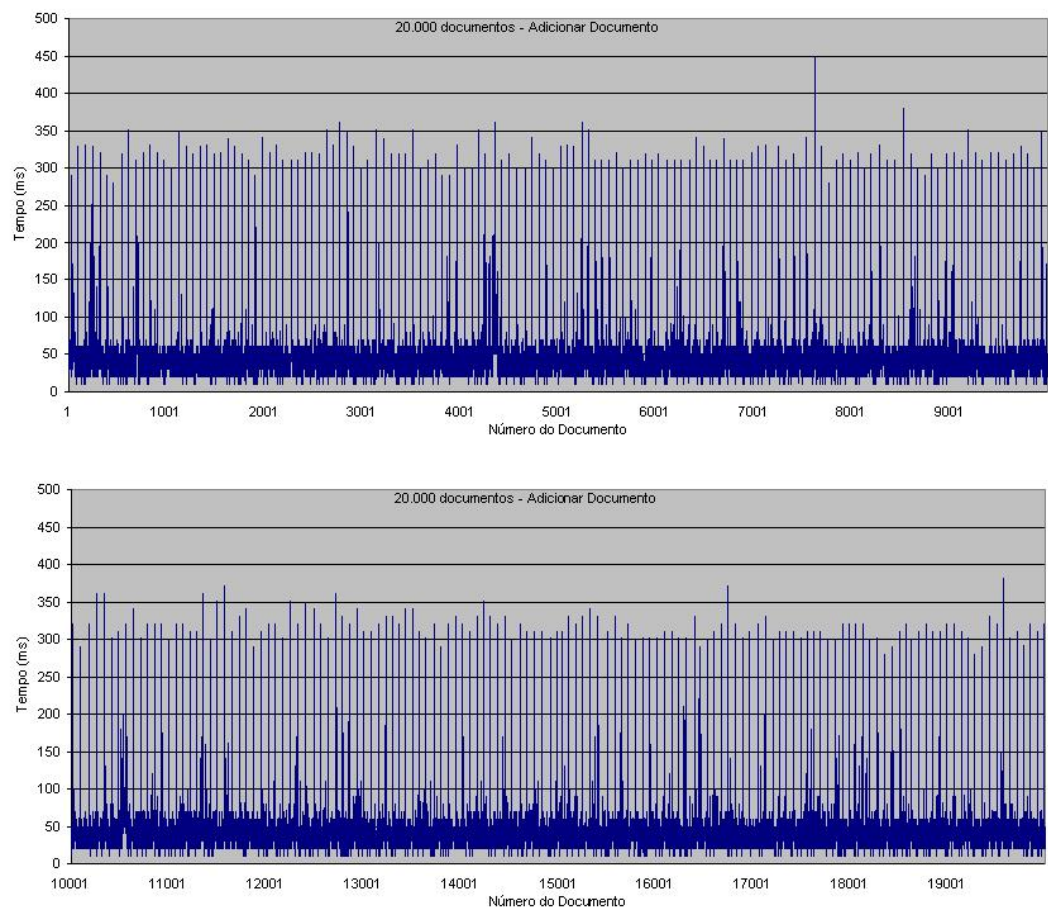


figura 48 - Gráficos do tempo despendido ao adicionar 20.000 documentos

<i>tempos registados (ms)</i>	
média	45
máximo	450

Tabela 6 - Tempos registados ao adicionar 20.000 documentos

E. 10.000 documentos

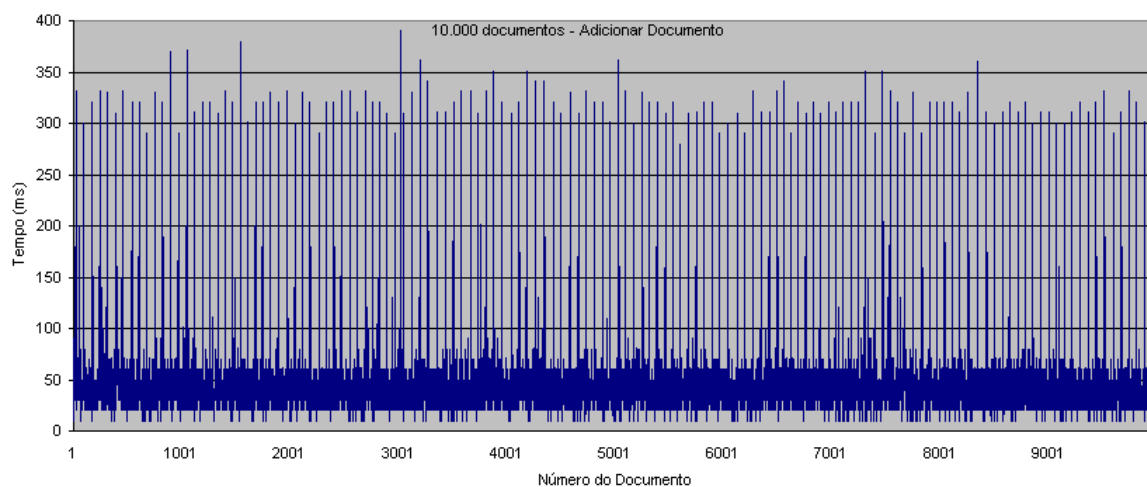


figura 49 - Gráfico do tempo despendido ao adicionar 10.000 documentos

<i>tempos registados (ms)</i>	
média	44
máximo	390

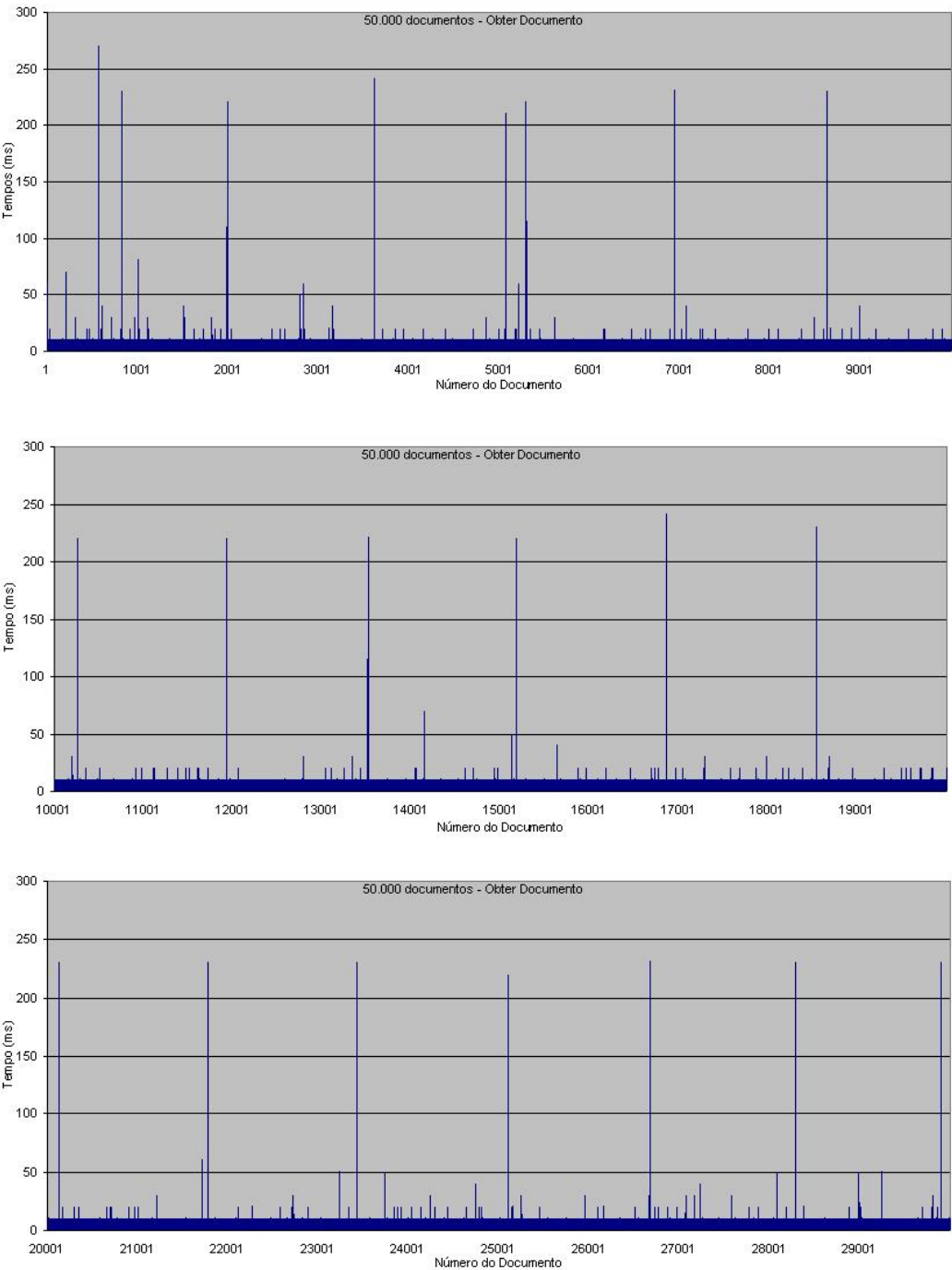
Tabela 7 - Tempos registados ao adicionar 10.000 documentos

Observando os gráficos e as tabelas de tempo constata-se que, em média, um documento demora cerca de 44 ms para ser adicionado à base de dados. Observou-se um fenómeno curioso que é a periodicidade em que um documento demora significativamente mais tempo a ser adicionado à base de dados *XML*. Como este aspecto não entra no âmbito deste estudo fica somente registado o acontecimento, mas não se apresenta nenhuma conclusão quanto à origem deste factor, uma vez que as causas podem ser diversas.

8.3.1.2 Obter Documentos

Apresentam-se de seguida os gráficos com o tempo despendido para obter um documento da base de dados *XML*. Apresenta-se também uma tabela com os tempos máximos e tempos médios para realizar esta operação.

A. 50.000 documentos



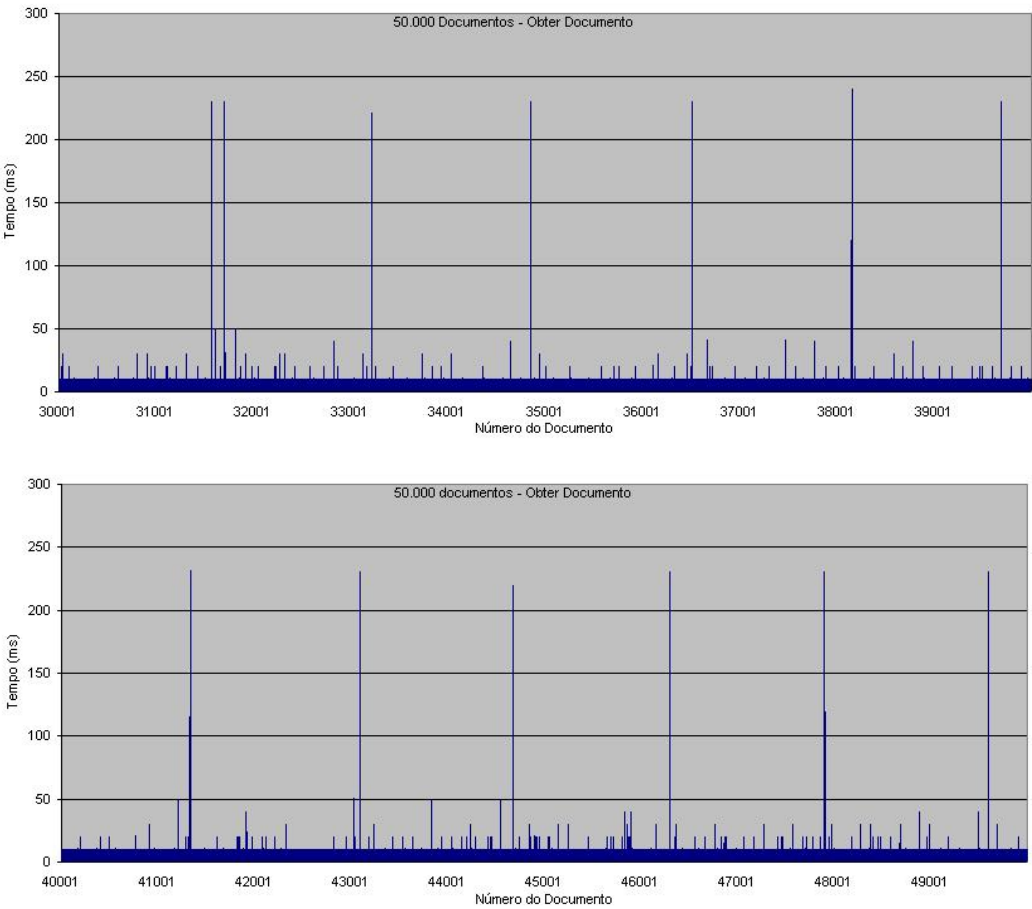
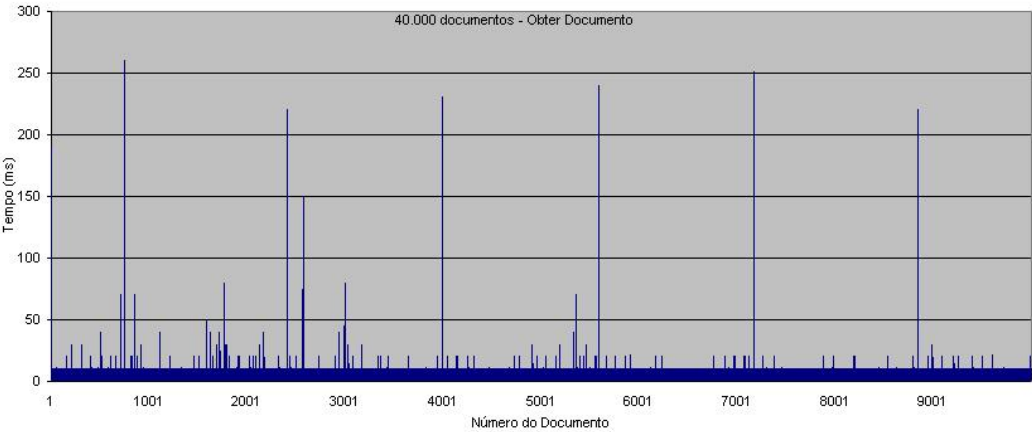


figura 50 - Gráficos do tempo despendido ao obter 50.000 documentos

<i>tempos registados (ms)</i>	
média	3
máximo	270

Tabela 8 - Tempos registados ao obter 50.000 documentos

A. 40.000 documentos



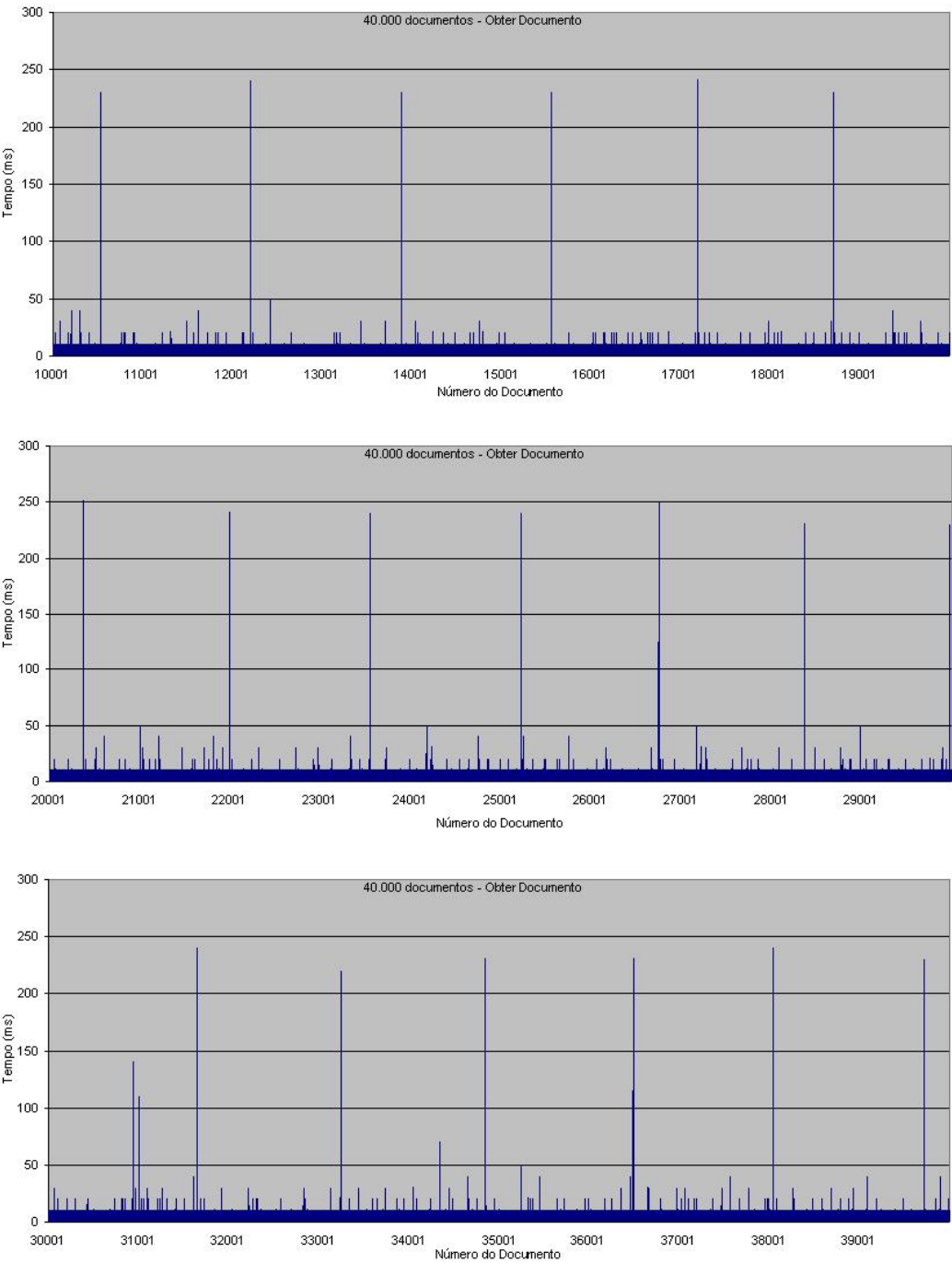


figura 51 - Gráficos do tempo despendido ao obter 40.000 documentos

<i>tempos registados (ms)</i>	
média	3
máximo	260

Tabela 9 - Tempos registados ao obter 40.000 documentos

B. 30.000 documentos

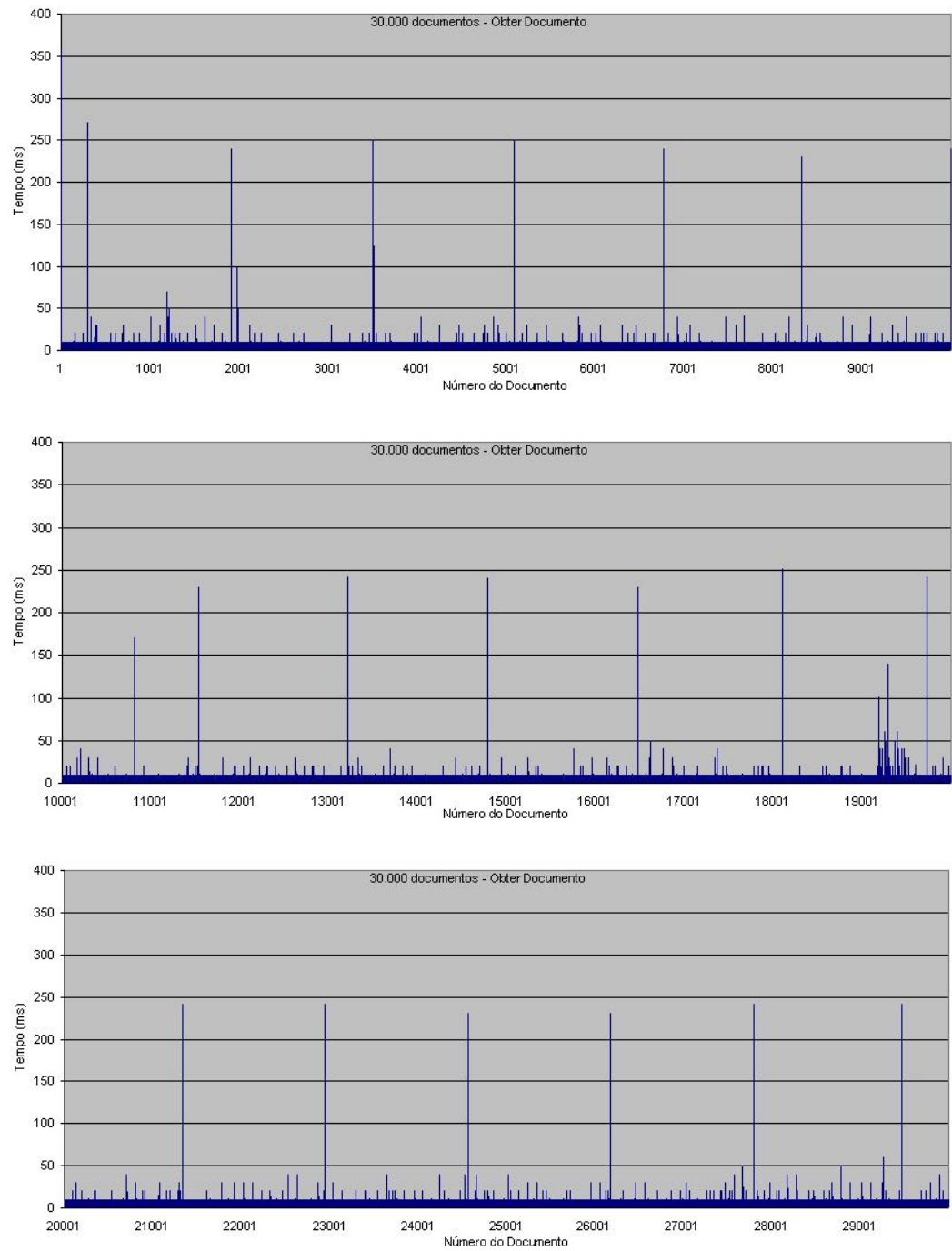


figura 52 - Gráficos do tempo despendido ao obter 30.000 documentos

<i>tempos registados (ms)</i>	
média	3
máximo	360

Tabela 10 - Tempos registados ao obter 30.000 documentos

C. 20.000 documentos

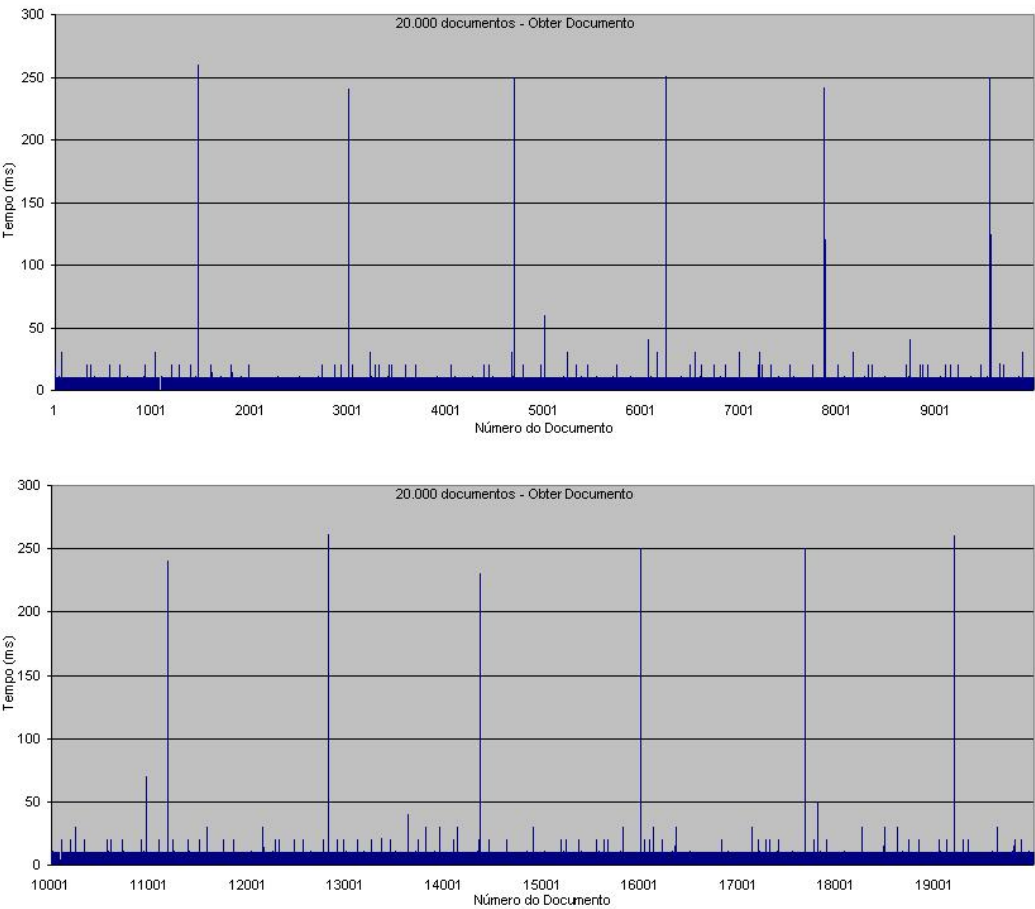


figura 53 - Gráficos do tempo despendido ao obter 20.000 documentos

<i>tempos registados (ms)</i>	
média	3
máximo	261

Tabela 11 - Tempos registados ao obter 20.000 documentos

D. 10.000 documentos

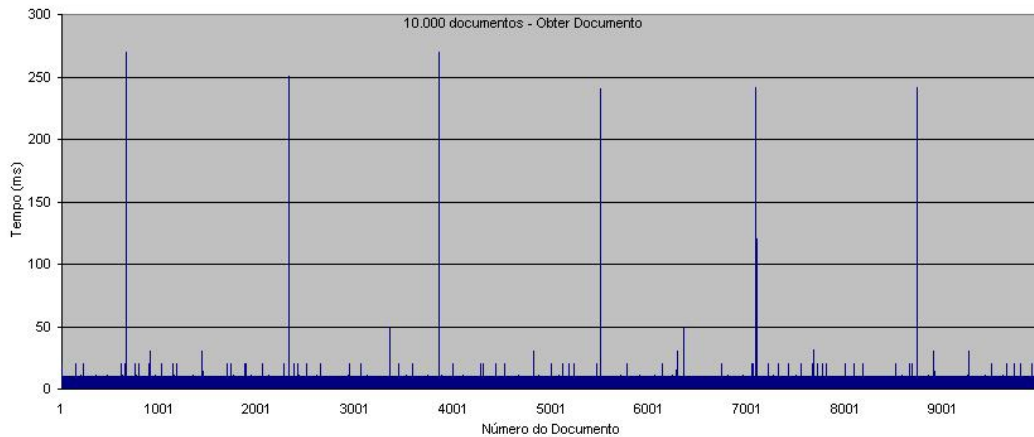


figura 54 - Gráfico do tempo despendido ao obter 10.000 documentos

<i>tempos registados (ms)</i>	
média	3
máximo	270

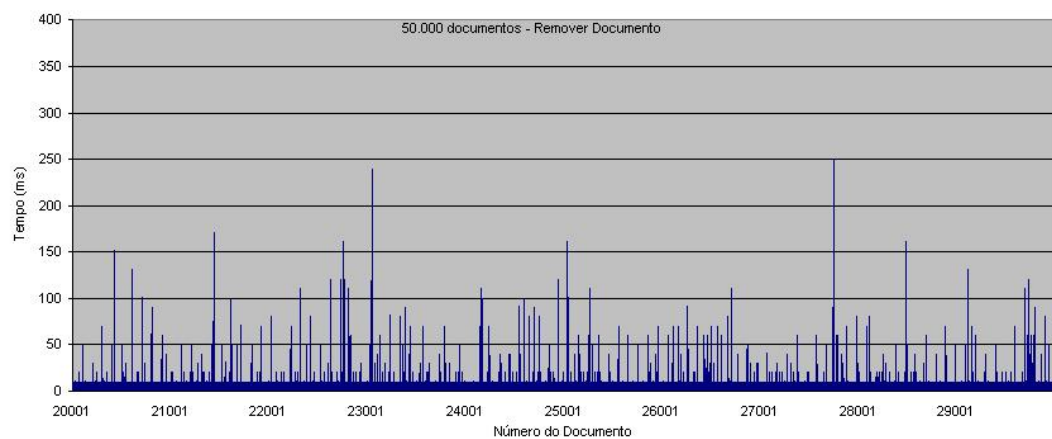
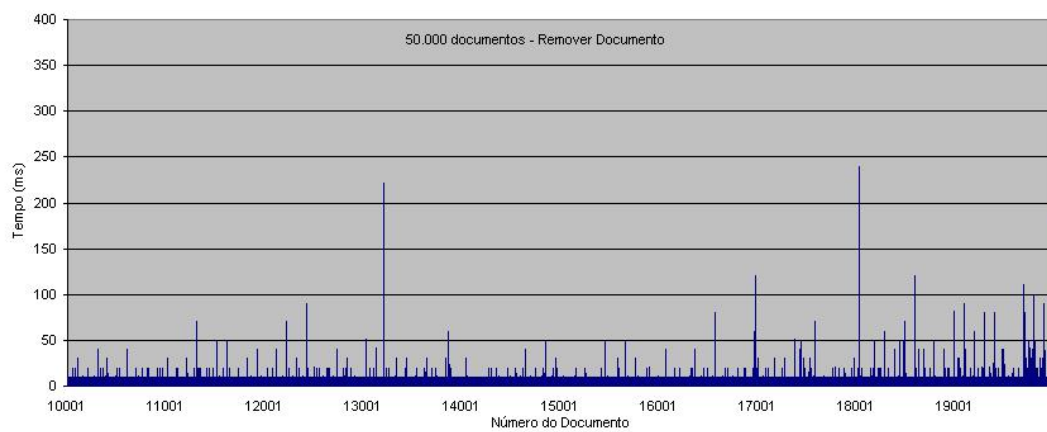
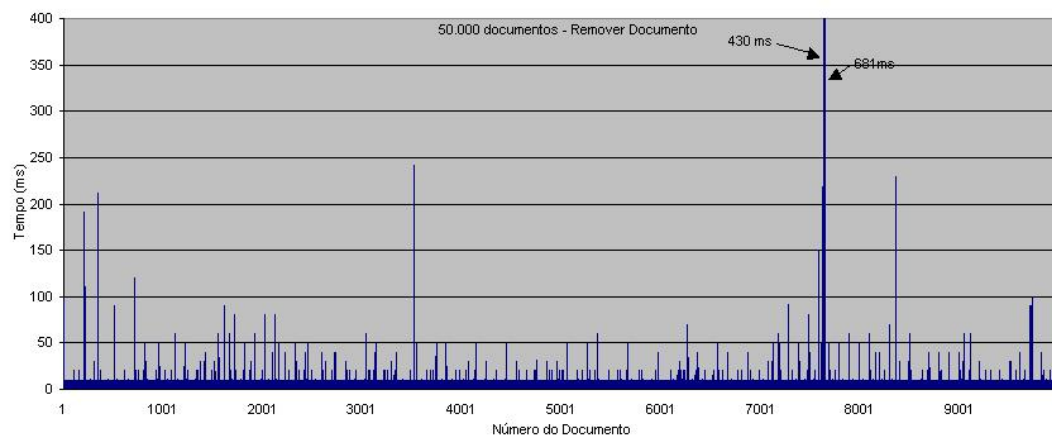
Tabela 12 - Tempos registados ao obter 10.000 documentos

Analisando os gráficos e as tabelas com os tempos conclui-se que a operação de obter um documento da base de dados *XML* é praticamente instantânea, demorando cerca de 3 ms , em média, para obter um documento, independentemente do número de documentos existentes na base de dados *XML*. Mais uma vez ocorre um fenómeno com uma certa periodicidade de um documento que demora significativamente mais tempo que os outros.

8.3.1.3 Remover Documentos

Apresentam-se de seguida os gráficos com os tempos registados para remover um documento da base de dados *XML*. Apresenta-se também uma tabela com os tempos máximos e tempos médios para realizar esta operação.

A. 50.000 documentos



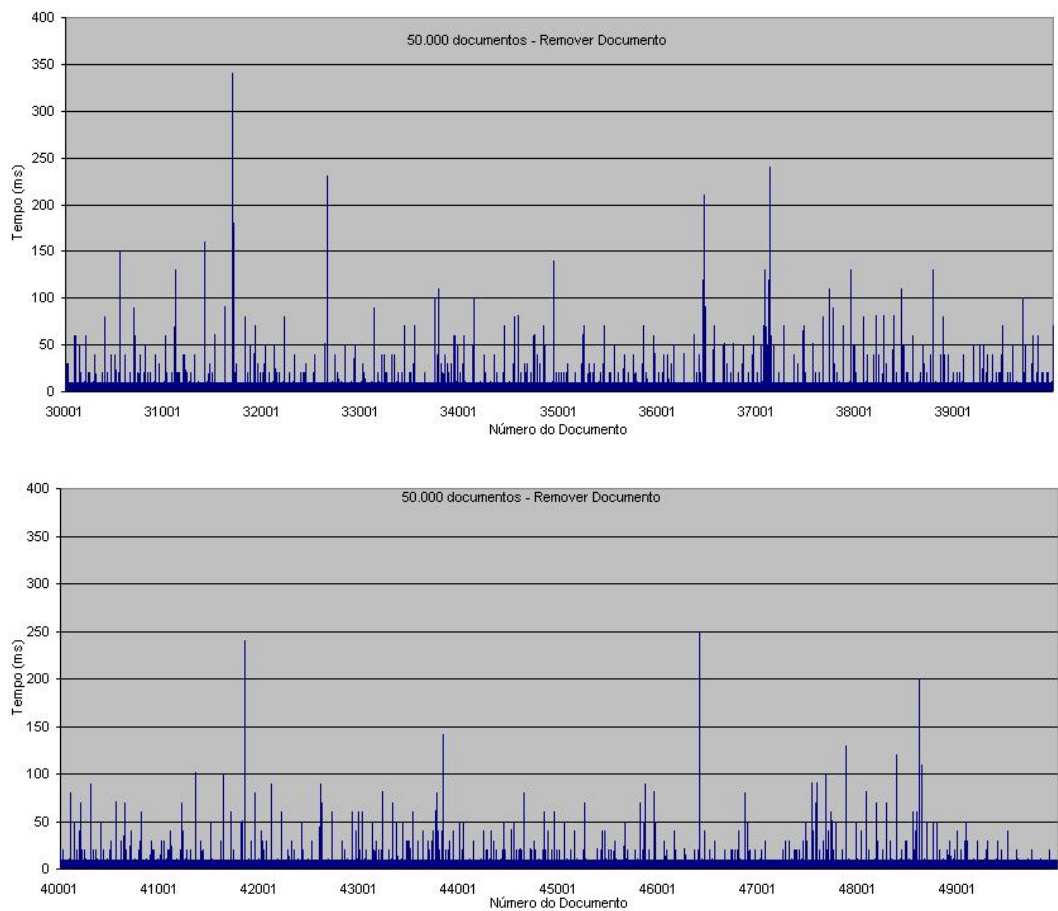
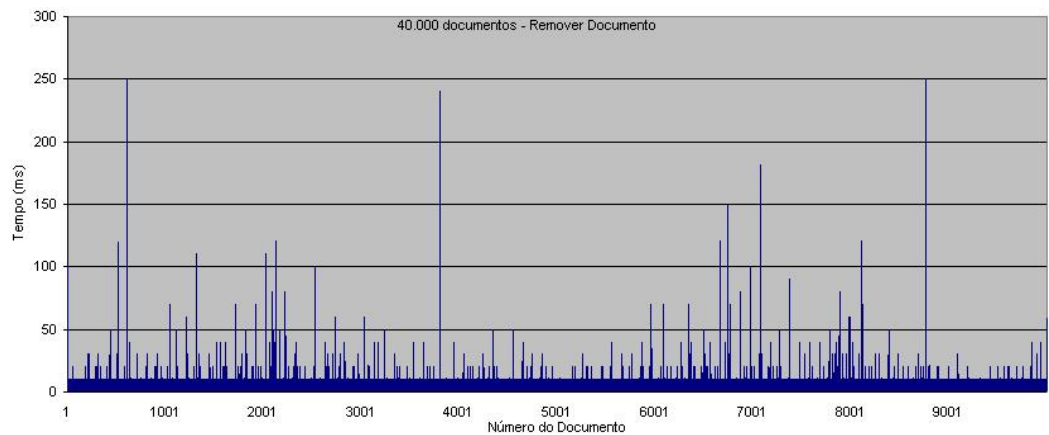


figura 55 - Gráficos do tempo despendido ao remover 50.000 documentos

<i>tempos registados (ms)</i>	
média	6
máximo	681

Tabela 13 - Tempos registados ao remover 50.000 documentos

A. 40.000 documentos



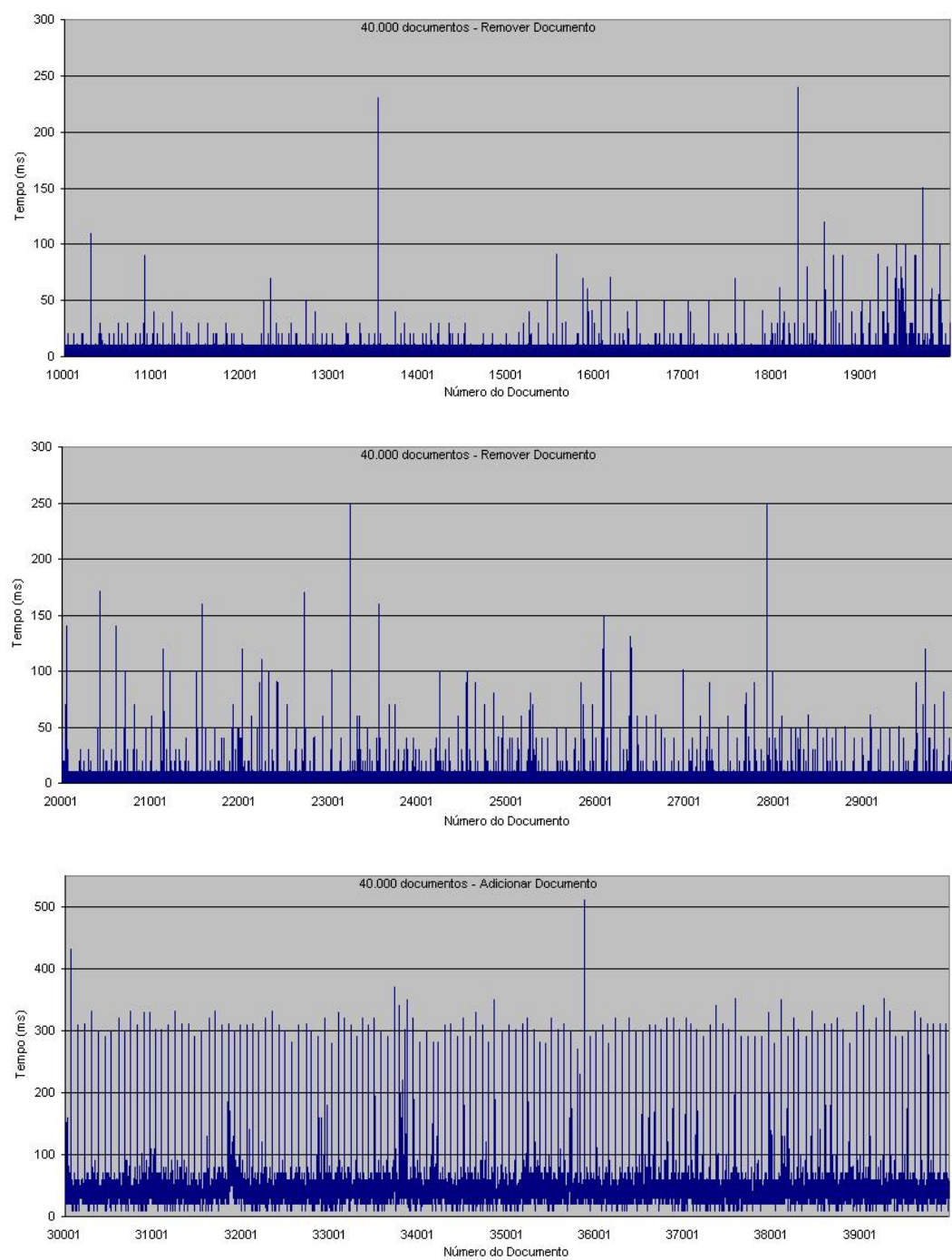


figura 56 - Gráficos do tempo despendido ao remover 40.000 documentos

<i>tempos registados (ms)</i>	
média	6
máximo	250

Tabela 14 - Tempos registados ao remover 40.000 documentos

B. 30.000 documentos

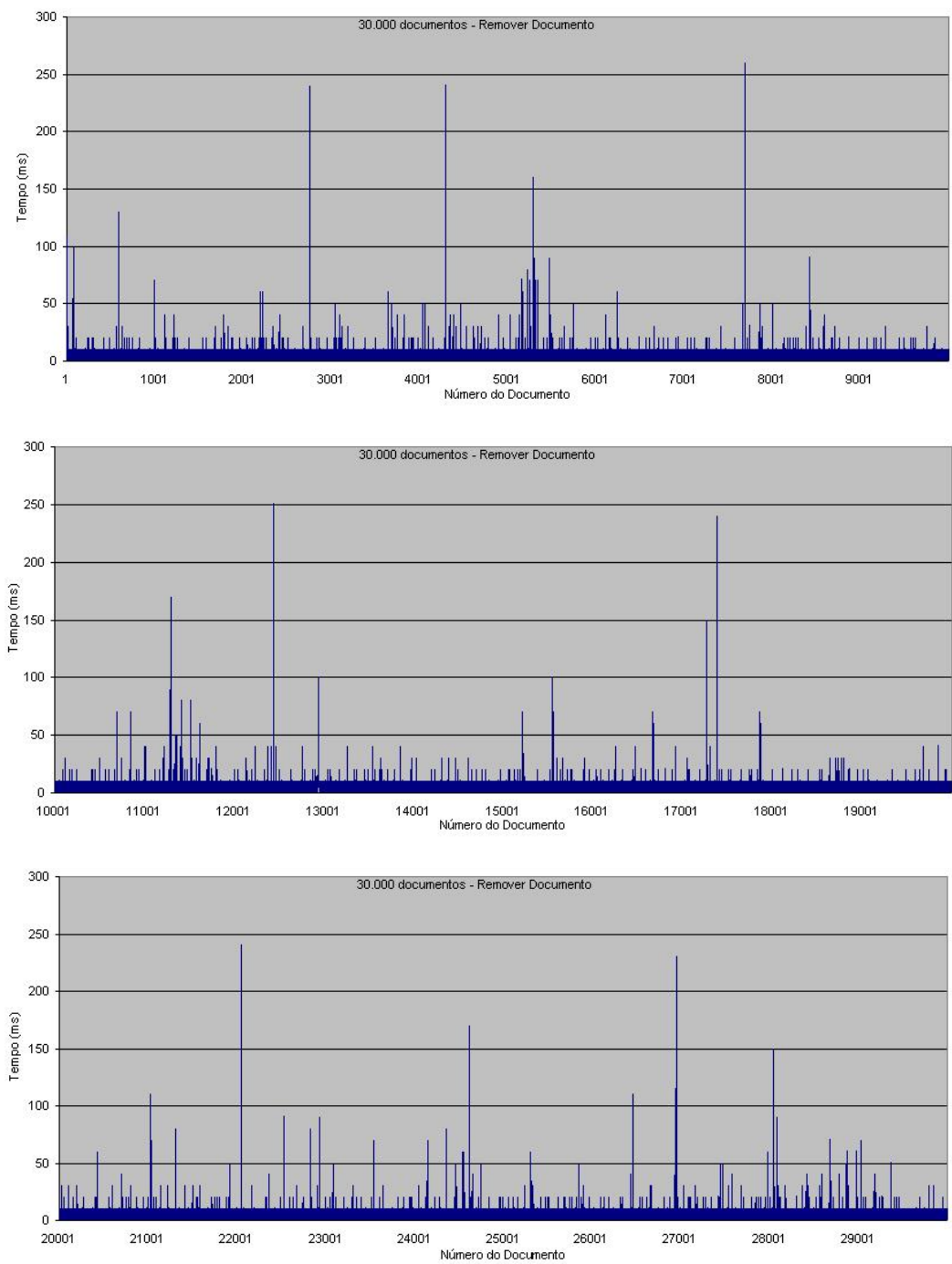


figura 57 - Gráficos do tempo despendido ao remover 30.000 documentos

<i>tempos registados (ms)</i>	
média	6
máximo	260

Tabela 15 - Tempos registados ao remover 30.000 documentos

C. 20.000 documentos

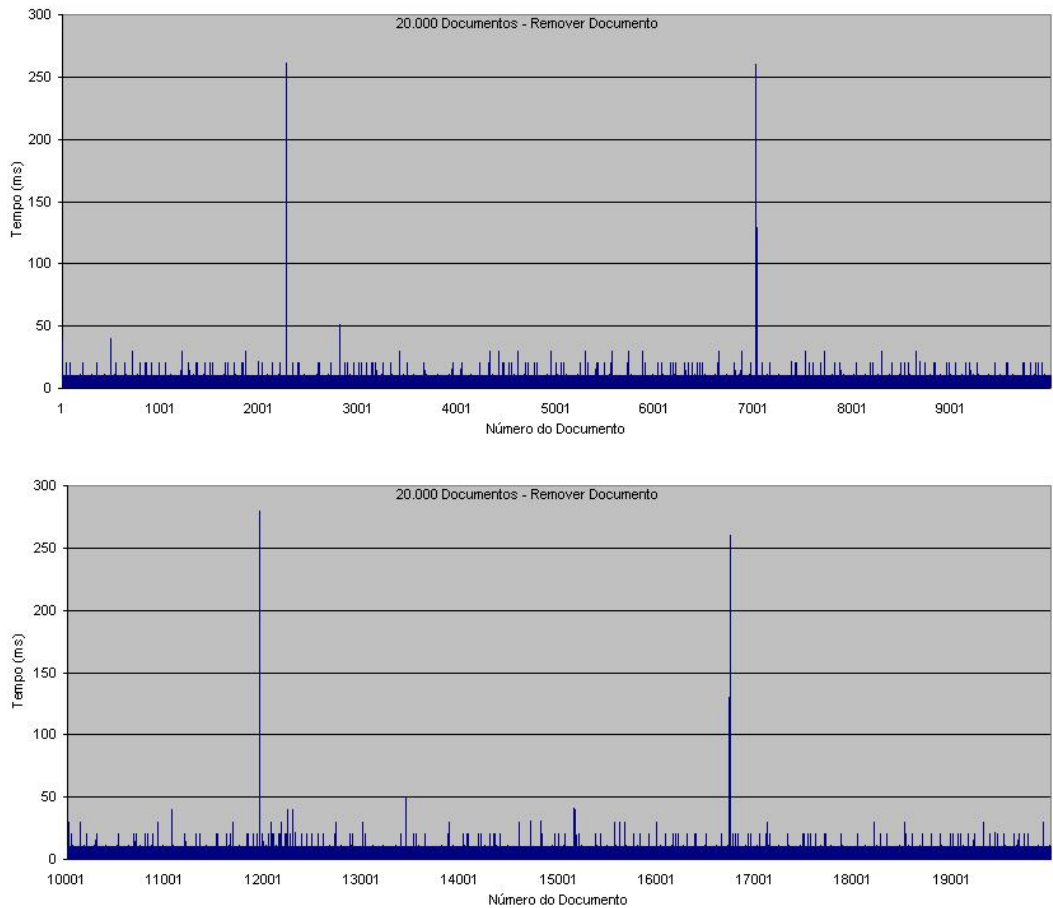


figura 58 - Gráficos do tempo despendido ao remover 20.000 documentos

<i>tempos registados (ms)</i>	
média	6
máximo	280

Tabela 16 - Tempos registados ao remover 20.000 documentos

D. 10.000 documentos

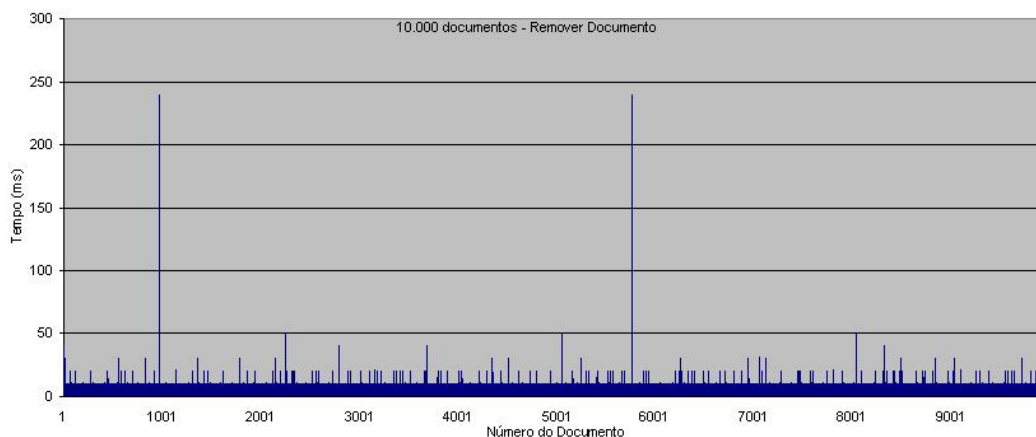


figura 59 - Gráfico do tempo despendido ao remover 10.000 documentos

<i>tempos registados (ms)</i>	
média	6
máximo	240

Tabela 17 - Tempos registados ao remover 10.000 documentos

Analisando os gráficos e as tabelas com os tempos conclui-se que a operação de remover um documento da base de dados *XML* demora, em média, cerca de 6 ms, independentemente do número de documentos existentes na base de dados *XML*.

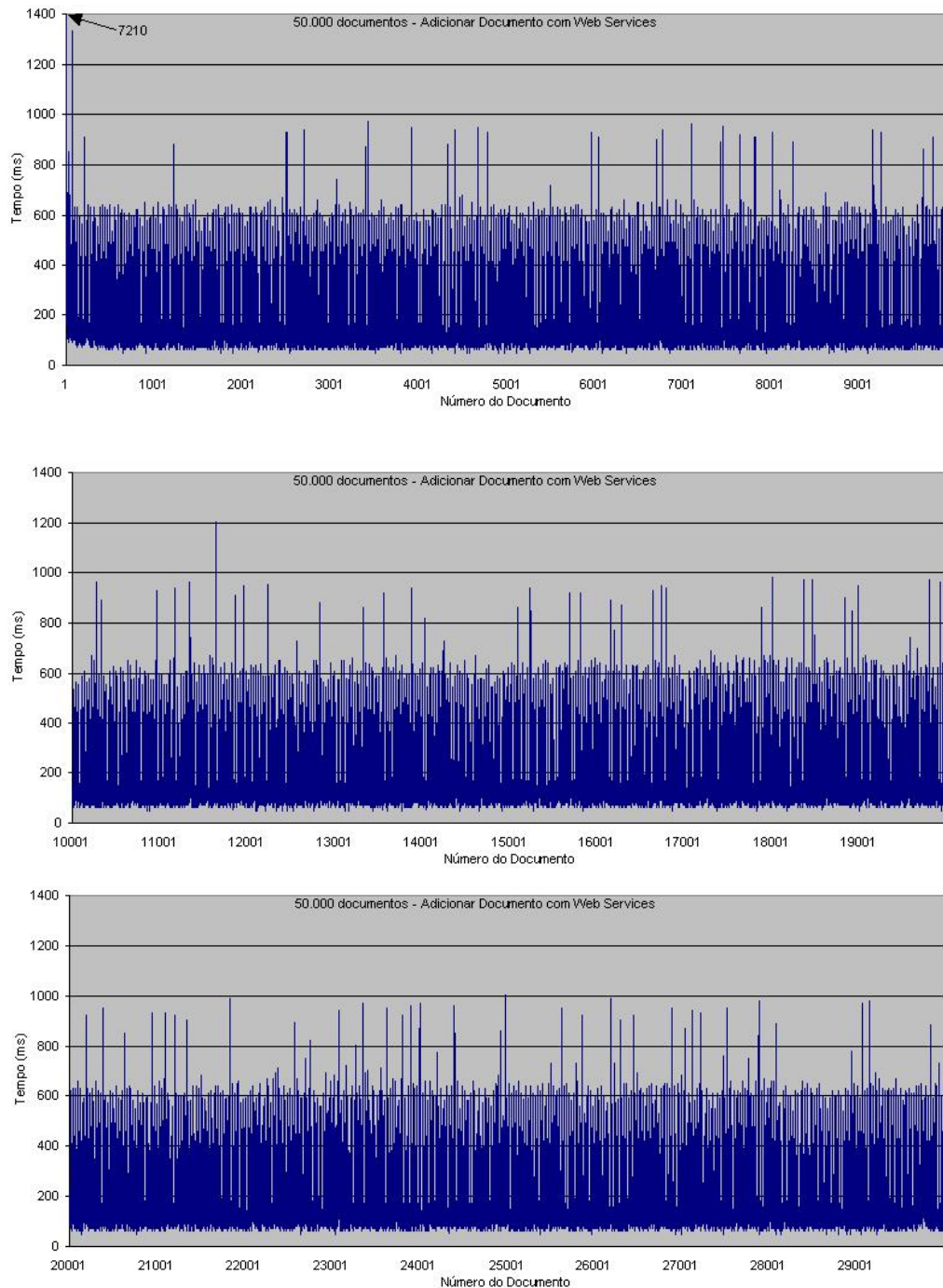
Pela análise dos gráfico apresentados na secção 8.3.1 pode concluir-se que a aplicação *XIndex* é bastante escalável, apresentando um comportamento idêntico nas operações de inserir, obter e remover um documento, independentemente do número de documentos que se encontram armazenados.

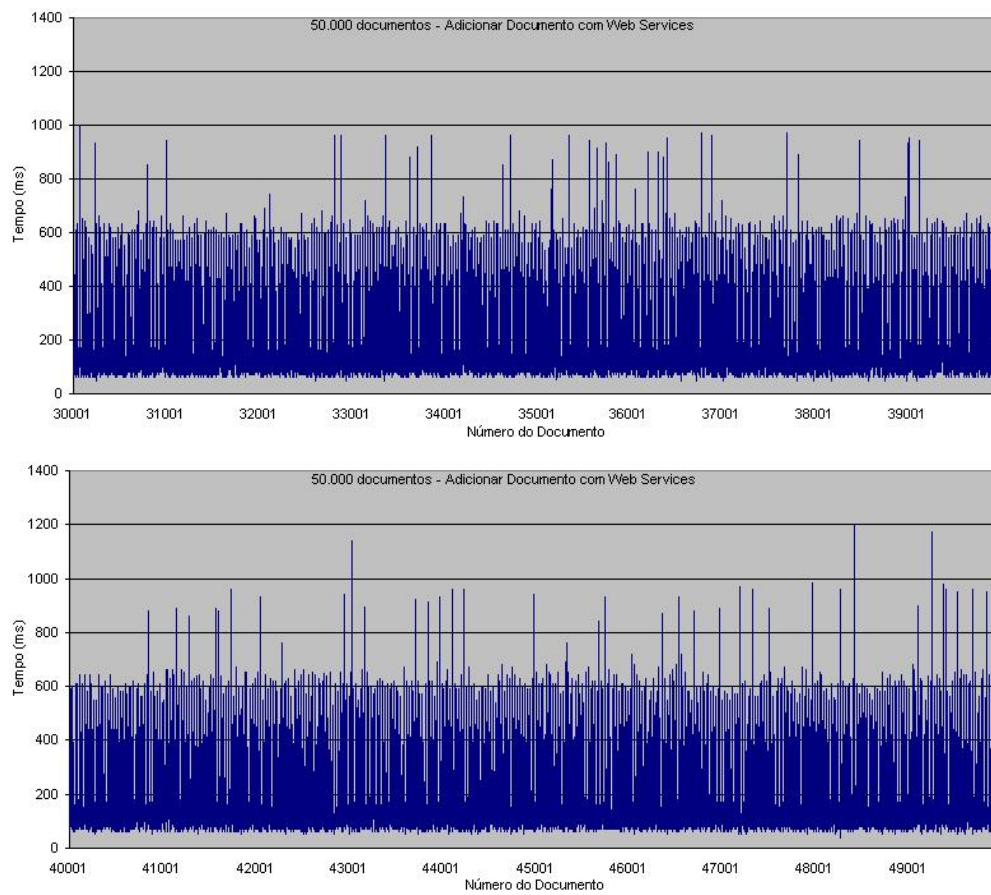
8.3.2 Acesso com *Web Services*

O tempos de acesso utilizando *Web Services* vão servir para avaliar qual o tempo de atraso que os *Web Services* criam no sistema. Como, de acordo com estudo anterior, se verifica que a base de dados *XML* apresenta um comportamento semelhante independentemente do número de documentos que armazena, somente se fez um teste para adicionar 50.000 documentos e registaram-se os tempos de inserção média de um documento.

Comparando este valor com o tempo médio obtido na secção 8.3.1.1 é possível obter uma estimativa para o atraso que o acesso com *Web Services* à base de dados *XML* provoca.

8.3.2.1 Adicionar 50.000 documentos com *Web Services*



figura 60 - Gráficos do tempo despendido com *Web Services*

<i>tempos registados (ms)</i>	
Média	157
Máximo	7210

Tabela 18 - Tempos registados com *Web Services*

Analisando os gráficos e a tabela com os tempos registados conclui-se que, em média, adicionar um documento à base de dados *XML* com *Web Services* demora cerca de 157 ms. Na secção 8.3.1.1 verificou-se que utilizando directamente o *driver XML:DB API* o tempo médio despendido era de 44 ms.

Pode então concluir-se que o acesso com *Web Services* causa um atraso de cerca de 110 ms no acesso à base de dados *XML*. Se compararmos este tempo de atraso com os tempos de pesquisa que se apresentam na secção 8.3.3 chegamos à conclusão que este atraso pode ser insignificante ou pode apresentar um agravamento significativo no tempo de resposta do sistema. No fim da próxima secção analisa-se este detalhe em mais pormenor.

8.3.3 Análise das pesquisas

Efectuaram-se testes, utilizando um acesso à base de dados *XML* através do *driver XML:DB API*, para medir a performance da base de dados *XIndex* quanto às funcionalidades de pesquisa.

Foram efectuadas pesquisas de palavras que aparecem nos descritores e pelo Id de um *AudioVisualSegment*. Pretende-se também analisar o desempenho dos índices criados pelo *XIndex*. Para tal, efectuaram-se pesquisas antes e depois de realizar a indexação dos elementos que são pesquisados.

Apresentam-se também os tempos que os índices demoram a ser criados e o tamanho que ocupam.

A. Pesquisas em 50.000 documentos

Descritor	Tempo pesquisa		Número ocorrências
	sem índices	com índices	
Agricultura	32m30s014ms	23m58s238ms	5058
Orçamento	18m38s458ms	16m56s893ms	20110
Pesca	41m58s261ms	19m57s361ms	20184

Tabela 19 - Tempos de pesquisa por descritores em 50.000 documentos

ID	Tempo pesquisa	
	sem índices	com índices
500	19m36s081ms	841ms
75000	19m46s196ms	531ms
350000	19m35s991ms	321ms

Tabela 20 - Tempos de pesquisa por ID em 50.000 documentos

Características da Base de Dados		
Índice	Tempo criação	Tamanho do Índice (MB)
ID	12m37s	440
Keyword	2h6m27s	427
Tamanho da Base de Dados (sem índices): 292 MB		

Tabela 21 - Características da base de dados XML com 50.000 documentos

B. Pesquisas em 40.000 documentos

Descritor	Tempo pesquisa		Número ocorrências
	sem índices	com índices	
Agricultura	19m51s203ms	20m09s740ms	4084
Orçamento	16m31s186ms	16m04s708ms	16045
Pesca	17m12s284ms	17m38s722ms	16061

Tabela 22 - Tempos de pesquisa por descritores em 40.000 documentos

ID	Tempo pesquisa	
	sem índices	com índices
500	15m17s820ms	1592ms
75000	15m16s808ms	421ms
350000	15m13s594ms	60ms

Tabela 23 - Tempos de pesquisa por ID em 40.000 documentos

Características da Base de Dados		
Índice	Tempo criação	Tamanho do índice (MB)
ID	10m26s	440
Keyword	1h21m35s	420
Tamanho da Base de Dados (sem índices): 292 MB		

Tabela 24 - Características da base de dados XML com 40.000 documentos

C. Pesquisas em 30.000 documentos

Descritor	Tempo pesquisa		Número ocorrências
	sem índices	com índices	
Agricultura	15m05s582ms	15m51s999ms	3098
Orçamento	13m36s174ms	13m22s433ms	11987
Pesca	14m03s703ms	14m10s894ms	12082

Tabela 25 - Tempos de pesquisa por descritores em 30.000 documentos

ID	Tempo pesquisa	
	sem índices	com índices
500	11m43s312ms	210ms
75000	11m47s477ms	341ms
200000	11m42s580ms	350ms

Tabela 26 - Tempos de pesquisa por ID em 30.000 documentos

<i>Características da Base de Dados</i>		
Índice	Tempo criação	Tamanho do Índice (MB)
ID	6m13s	293
Keyword	1h2m3s	280
Tamanho da Base de Dados (sem índices): 194 MB		

Tabela 27 - Características da base de dados XML com 30.000 documentos

D. Pesquisas em 20.000 documentos

Descritor	<i>Tempo pesquisa</i>		Número ocorrências
	sem índices	com índices	
Agricultura	9m03s031ms	9m33s715ms	2073
Orçamento	9m52s822ms	9m14s587ms	7965
Pesca	9m48s155ms	9m57s319ms	8103

Tabela 28 - Tempos de pesquisa por descritores em 20.000 documentos

ID	<i>Tempo pesquisa</i>	
	sem índices	com índices
500	7m21s114ms	220ms
75000	7m14s534ms	391ms
150000	7m14s786ms	80ms

Tabela 29 - Tempos de pesquisa por ID em 20.000 documentos

<i>Características da Base de Dados</i>		
Índice	Tempo criação	Tamanho do Índice (MB)
ID	4m22s	195
Keyword	35m47s	186
Tamanho da Base de Dados (sem índices): 129 MB		

Tabela 30 - Características da base de dados XML com 20.000 documentos

E. Pesquisas em 10.000 documentos

Descritor	<i>Tempo pesquisa</i>		Número ocorrências
	sem índices	com índices	
Agricultura	4m26s934ms	4m31s010ms	1028
Orçamento	5m00s172ms	5m02s595ms	3927
Pesca	5m01s984ms	5m06s251ms	4085

Tabela 31 - Tempos de pesquisa por descritores em 10.000 documentos

ID	<i>Tempo pesquisa</i>	
	sem índices	com índices
500	3m39s156ms	220ms
37500	3m38s784ms	231ms
75000	3m38s665ms	70ms

Tabela 32 - Tempos de pesquisa por ID em 10.000 documentos

<i>Características da Base de Dados</i>		
Índice	Tempo criação	Tamanho do Índice (MB)
ID	1m54s	87
Keyword	15m27s	83
Tamanho da Base de Dados (sem índices) : 87 MB		

Tabela 33 - Características da base de dados XML com 10.000 documentos

Analisando as tabelas constata-se que, mesmo com um número reduzido de documentos, os tempos de pesquisa sem indexação são demasiado elevados. O tempo de pesquisa quer por um descritor ou pelo *Id* de um *AudioVisualSegment* demora sempre mais do que três minutos.

Utilizando indexação do atributo *Id* verifica-se que os tempos de pesquisa melhoram significativamente. Em geral qualquer que seja o número de documentos armazenados na base de dados obtêm-se tempos de pesquisa inferiores a um segundo, o que é um valor aceitável.

O problema surge na indexação de texto nos descritores. Observando as tabelas com os tempos de pesquisa chega-se à conclusão que, muitas vezes, o índices não tem praticamente qualquer efeito sobre os tempos de pesquisa.

Outro aspecto que se deve ter em consideração é o tamanho que os índices ocupam em relação ao tamanho da base de dados com os documentos.

Quando existem muitos documentos na base de dados a tendência do índice é ocupar o dobro do espaço que ocupa a base de dados.

Quanto ao efeito do atraso provocado pelos *Web Services* no desempenho do sistema, no caso das pesquisas por descritores cujos tempos são sempre superiores a três minutos, não tem qualquer influência no desempenho do sistema. No entanto no caso das pesquisas por um *ID* com indexação deste elemento os *Web Services* já provocam um atraso considerável, chegando mesmo por vezes a ser superior ao tempo de pesquisa.

9 CONCLUSÕES E TRABALHO FUTURO

A construção de um sistema que permita a descrição de conteúdo audiovisual pode tornar-se uma tarefa muito complexa. É necessário estudar bem quais os elementos que se pretendem descrever pois o *standard MPEG-7* disponibiliza um leque de elementos bastante extenso para descrever os aspectos relacionados com o conteúdo e organização dos vários tipos de media existentes. Ainda não existem muitas ferramentas disponíveis para este tipo de aplicações e torna-se necessário avaliar os *standards* e tecnologias disponíveis para realizar este tipo de tarefas.

O *MPEG-7* respondeu plenamente às necessidades de indexação e caracterização dos vídeos.

Utilizando bases de dados nativas *XML*, espera-se que o desempenho do sistema na parte de indexação e pesquisa de informação seja feito de uma forma mais eficiente, tendo em conta a própria natureza dos documentos a armazenar.

A utilização de *Web Services* na camada lógica permite criar um nível de abstracção entre a camada de dados e a camada de apresentação. Uma abordagem deste género permite aumentar bastante a escalabilidade do sistema de arquivo audiovisual. Verificou-se que os *Web Services* permitem também criar facilmente interoperabilidade entre aplicações desenvolvidas em diferentes plataformas com diferentes linguagens de programação.

O único componente que não respondeu completamente às necessidades do sistema foi o *SGBD* da base de dados *XML*. O *XIndex* apresentou bastante limitações nas pesquisas por texto livre e o seu sistema de indexação mostrou-se ineficaz para resolver este problema.

A nível de trabalho futuro é necessário estudar outros *SGBD's XML* que permitam obter boas performances em pesquisas por texto livre. É também necessário melhorar a apresentação da interface *Web* de visualização dos vídeos.

REFERÊNCIAS

- [ANSI, 1986] ANSI, American National Standards Institute, "*The Database Language SQL*", Documento ANSI X3.135, 1986
- [APACHE, 2003a] Apache . "*Apache XIndex*", <http://xml.apache.org/xindice/>, Março 2003
- [APACHE, 2003b] Apache . "*The Jakarta Site – Apache Tomcat*", <http://jakarta.apache.org/tomcat/>, Janeiro 2003
- [APACHE, 2003c] Apache . "*Apache Axis*", <http://ws.apache.org/axis/>, Janeiro 2003
- [APACHE, 2003d] Apache, "*Axis Reference Guide*", <http://ws.apache.org/axis/java/reference.html>, Maio 2003
- [ARMSTRONG, 2003] Armstrong, Eric. et al , "*The Java Web Services Tutorial* ", Fevereiro 2003
- [BOURETT, 2003] Bourett, Ronald. "*XML and Databases*" , <http://www.rpbouret.com/xml/XMLAndDatabases.htm>, Janeiro 2003.
- [BOX, 1998] Box, Don. "*Essential COM.*" Addison-Wesley Publishing Company, ISBN 0-201-63446-5, 1998
- [DIX, 1998] Dix, A., J. Finlay, G. Abowd, e R. Beale. "*Human-Computer Interaction*". Prentice Hall, 2.^a edição, 1998
- [DSTC, 2003a] DSTC, "*XMLdbGUI - Download*" , <http://titanium.dstc.edu.au/xml/xmldbgui/download.shtml>, Março 2003
- [DSTC, 2003b] DSTC, "*XMLdbGUI - Overview of XMLdbGUI*" , <http://titanium.dstc.edu.au/xml/xmldbgui/index.shtml>, Março 2003
- [European, 2003] European Communities, "*Eurovoc thesaurus*", <http://europa.eu.int/celex/eurovoc/>, Abril 2003
- [ISO, 2001] ISO, "*Standard Generalized Markup Language (SGML)*", ISO 8879:1986, Agosto 2001
- [LAUX, 2000] Laux, Andreas. et al "*XML:DB Initiative: XUpdate - XML Update Language*", XML:DB Initiative for XML Databases, Setembro 2000, <http://www.xmldb.org/xupdate/xupdate-wd.html>, consultado em Março 2003
- [MARTINEZ, 2002] Martínez, José M. , "*MPEG-7 Overview (version 8.0)*", ISO/IEC, Julho 2002
- [MICROSOFT, 1995] Microsoft , "*The Component Object Model Specification*", 1995, <http://www.microsoft.com/com/resources/specs.asp>, consultado em Abril 2003
- [MICROSOFT, 2003a] Microsoft, "*Microsoft Open Database Connectivity (ODBC)*", <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/hm/odbcstartpage1.asp>, Junho 2003
- [MICROSOFT, 2003b] Microsoft, "*ODBC Overview*", <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/odbc/hm/odch01pr.asp>, Junho 2003

- [MICROSOFT, 2003c] Microsoft, “*Microsoft Visual C# - An innovative language and tool for building .NET-connected solutions*”, <http://msdn.microsoft.com/vcsharp/>, Abril 2003
- [MICROSOFT, 2003d] Microsoft, “*Microsoft Visual Studio Rapidly build applications for Microsoft Windows, Web and mobile devices*”, <http://msdn.microsoft.com/vcsharp/>, Abril 2003
- [ORACLE, 2003] Oracle Technology Network, “*WSDL Overview*”, <http://otn.oracle.com/tech/webservices/htdocs/wsvsm/wsdlover.html>, Junho 2003
- [PINTO, 2001a] Pinto, Joaquim Sousa. Almeida, Pedro. “*Diário da Assembleia da República Electrónico*”, Revista Departamento Engenharia Electrónica e Telecomunicações, ISSN: 1645-0493, Vol.3, N.º3, p. 209-213, Janeiro 2001
- [PINTO, 2001b] Pinto , Joaquim Sousa et. al. , “*Portuguese Parliamentary Records Digital Library*” , In Ahmed K. Elmagarmid , William J. McIver Jr, “*The Ongoing March Toward Digital Government*”, Computer, Vol. 34, N.º 2, p. 38, IEEE Computer Society, Fevereiro 2001
- [RYU, 2002] Ryu et al. , “*MPEG-7 Authoring Tool*”, ACM International Multimedia Conference, Proceedings of the tenth ACM international conference on Multimedia, p. 267-270, ISBN:1-58113-620-X , 2002
- [SALEMBIER, 2001] Salembier, Philippe , Smith, John R. , “*MPEG-7 Multimedia Description Schemes*”, IEEE Vol. 11 N.º 6 , Junho 2001
- [SRINIVASAN, 1995] Srinivasan, R., Sun Microsystems. “*RFC 1831 - RPC: Remote Procedure Call Protocol Specification Version 2*”, Network Working Group, Agosto 1995
- [SUN, 2003a] Sun Microsystems, “*Java Architecture for XML Binding (JAXB)*”, <http://java.sun.com/xml/jaxb/> , Março 2003
- [SUN, 2003b] Fordin, Scott, “*The Java™ Architecture for XML Binding (JAXB) User’s Guide*”, Sun Microsystems , documentação incluída no pacote *Java Web Services Developer Pack*, Fevereiro 2003
- [SUN, 2003c] Sun Microsystems, “*Sun ONE Studio*”, <http://www.sun.com/software/sundev/index.html>, Fevereiro 2003
- [SUN, 2003d] Sun Microsystems, “*JDBC™ 4.0 API Specification*”, <http://jcp.org/en/jsr/detail?id=221>, Junho 2003
- [SUN, 2003e] Sun Microsystems, “*JDBC™ 4.0 Data Access API*”, <http://java.sun.com/products/jdbc/>, Junho 2003
- [SUN, 2003f] Sun Microsystems, “*Java Media Framework API*”, <http://java.sun.com/products/java-media/jmf/>, Junho 2003
- [SUN, 2003g] Sun Microsystems, “*Java 2 Platform, Standard Edition (J2SE)*”, <http://java.sun.com/j2se/>, Junho 2003
- [W3C, 1999a] W3C, “*HTML 4.01 Specification*”, <http://www.w3c.org/TR/html4>, Dezembro 1999
- [W3C, 1999b] W3C, “*XML Path Language (XPath) Version 1.0*”, <http://www.w3.org/TR/xpath>, Novembro 1999

- [W3C, 2000a] W3C . “*Extensible Markup Language (XML)*” , <http://www.w3.org/TR/REC-xml>, Outubro 2000
- [W3C, 2000b] W3C . “*Datatypes for DTDs (DT4DTD) 1.0*”, <http://www.w3.org/TR/dt4dtd>, Janeiro 2000
- [W3C, 2001a] W3C . “*XML Schema Part 0: Primer*”, <http://www.w3.org/TR/xmlschema-0/>, Maio 2001
- [W3C, 2001b] W3C, “*Web Services Description Language (WSDL) 1.1*”, <http://www.w3.org/TR/wsdl>, Março 2001
- [W3C, 2002] W3C . “*Web Services Architecture Requirements*” , <http://www.w3.org/TR/wsa-reqs>, Novembro 2002
- [W3C, 2003] W3C . “*Simple Object Access Protocol (SOAP) 1.1*” , <http://www.w3.org/TR/SOAP/>, Janeiro 2003
- [W3SCHOLLS, 2003] W3Scholls, “*XML Schemas - Why?*” , http://www.w3schools.com/schema/schema_why.asp , Janeiro 2003

BIBLIOGRAFIA

- Anderson, R. et al , “*Professional ASP.NET 1.0*”, Wrox , ISBN: 0-7645-4396-1, 2001
- Basiura R. et al , “*Professional ASP.NET Web Services*”, Wrox , ISBN: 1-861005-45-8, 2001
- Day, Neil. , “*MPEG-7 Projects and Demos*”, ISO/IEC, Março 2001
- Graves, A. P., Lalmas, M. “*Video retrieval using an MPEG-7 based inference network*”, 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 339-346, Tampere, Finland, Agosto 2002.
- Liberty, J., Hurwitz, D. , “*Programming ASP. NET* ”, O’ Reilly Windows, ISBN: 0-5960-0171-1, 2002
- Pereira, F., “*MPEG-7 Requirements Document V.15*”, ISO/IEC, Julho 2001
- Rehm, Eric , “*Representing Internet Streaming Media Metadata using MPEG-7 Multimedia Description Schemes*”, <http://www.acm.org/sigs/sigmm/MM2000/ep/rehm/>, Outubro 2002
- Sybex Inc. , “*C# Complete* “, SYBEX , ISBN: 0-7821-4203-6, 2003
- Sybex Inc. , “*Java 2, j2se 1.4 Complete* “, SYBEX , ISBN: 0-7821-4102-1, 2003

ANEXO A

UTILIZAÇÃO DA FERRAMENTA JAXB

O primeiro passo necessário é instalar a plataforma *JAXB*. Neste caso em particular foi instalada na directoria `c:\jaxb`.

PROCEDIMENTO DE COMPILAÇÃO DO *XML SCHEMA*:

Inicialmente é necessário registar as seguintes variáveis de ambiente: **JAXB_HOME** e **JAXB_LIBS**:

Para registar as avariáveis deve-se abrir uma janela de *DOS* e efectuar os seguintes comandos:

```
set JAXB_HOME=<directoria de instalação do JAXB>
set JAXB_LIBS=%JAXB_HOME%\lib
```

Após efectuar o procedimento anterior deve-se fazer a actualização da variável de ambiente *CLASSPATH*, de modo a esta incluir as bibliotecas do *JAXB*. Este procedimento é efectuado executando o seguinte procedimento ¹:

```
set CLASSPATH=%CLASSPATH%;%JAXB_LIBS%\jaxb-api.jar;
%JAXB_LIBS%\jaxb-ri.jar;%JAXB_LIBS%\jaxb-xjc.jar;
%JAXB_LIBS%\jaxb-lib.jar;%JAXB_LIBS%\jaxp-api.jar;
%JAXB_LIBS%\xercesImpl.jar;%JAXB_LIBS%\sax.jar;
%JAXB_LIBS%\dom.jar;%JAXB_LIBS%\jax-qname.jar;
```

Após estes procedimentos deve mudar-se para a directoria onde se encontra o *XML Schema* a compilar. Neste caso o *XML Schema* encontrava-se na seguinte directoria:

```
cd %JAXB_HOME%\video
```

¹ Ter em atenção que este comando não tem quebras de linha

Em seguida deve utilizar-se o comando *xjc.bat* para gerar as classes *Java* a partir do *XML Schema*.

```
%JAXB_HOME%\bin\xjc.bat video.xsd -p ieeta.video.mpeg7
```

video.xsd é o nome do *XML Schema* que vai ser compilado por o *JAXB Binding Compiler*. O switch *-p ieeta.video.mpeg7* indica ao compilador *JAXB* para colocar as classes *Java* que vão ser geradas num *Java package* com o nome *ieeta.video.mpeg7*.

Finalmente deve abrir-se um novo projecto com o *SunONE Studio* para compilar as classes *Java* geradas pelo *JAXB*.

De seguida descrevem-se os passos a efectuar:

1. Criar um novo projecto com o nome *Video*
2. No *Filesystems* do projecto fazer o *mount* da directoria *c:\jaxb\Video*
3. Fazer a montagem dos ficheiros *JAR* que se encontram na directoria *c:\jaxb\lib*
4. Compilar as classes que se encontram dentro da directoria *ieeta*. (Utilizar o comando *Compile All*).

Quando se pretende construir uma aplicação para manipular documentos *XML* que estejam de acordo com o *XML Schema* compilado basta incluir no programa o *package ieeta.video.mpeg7* que contém todas as classes *JAVA* que permitem manipular informação *XML*.

Ao analisar o conteúdo da directoria *ieeta.video.mpeg7* observa-se que foram geradas um série de classes para os vários tipos de elementos definidos no *XML Schema* que foi compilado.

Na figura 61 pode observar-se o aspecto do explorador do projecto de teste com o nome *Video* depois de fazer a montagem dos ficheiros *JAR* e das directorias que contêm as classes geradas por o *JAXB Binding Compiler*.

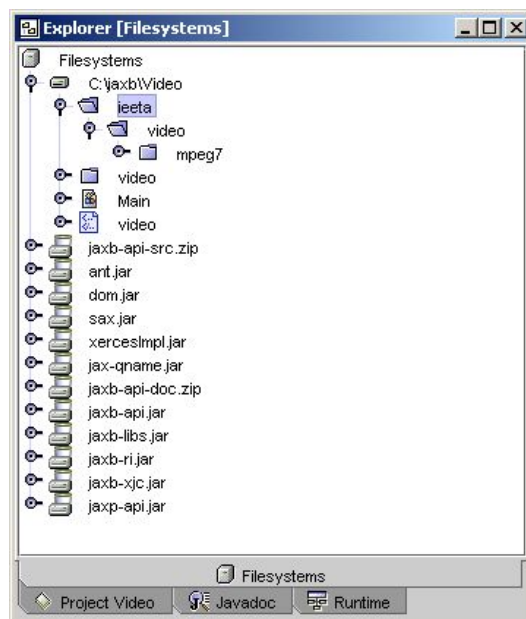


figura 61 - Explorador do projecto Video